RICE UNIVERSITY

Fluid–Structure Interaction Modeling of Modified-Porosity Parachutes and Parachute Clusters

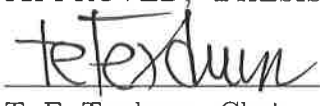
by

Joseph J. Boben, 2nd Lt, USAF

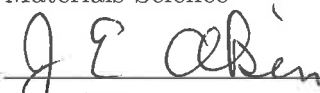
A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Science

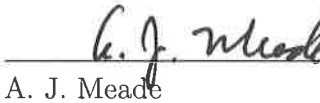
Approved, Thesis Committee:



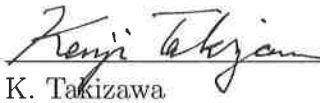
T. E. Tezduyar, Chair
Professor of Mechanical Engineering and
Materials Science



J. E. Akin
Professor of Mechanical Engineering and
Materials Science and Professor of
Computational and Applied Mathematics



A. J. Meade
Professor of Mechanical Engineering and
Materials Science



K. Takizawa
Associate Professor in Department of
Modern Mechanical Engineering and
Waseda Institute for Advanced Study
Waseda University, Tokyo, Japan

Houston, Texas
April 2013

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

Abstract

Fluid–Structure Interaction Modeling of Modified-Porosity Parachutes and Parachute Clusters

by

Joseph J. Boben

To increase aerodynamic performance, the geometric porosity of a ringsail spacecraft parachute canopy is sometimes increased, beyond the "rings" and "sails" with hundreds of "ring gaps" and "sail slits." This creates extra computational challenges for fluid–structure interaction (FSI) modeling of clusters of such parachutes, beyond those created by the lightness of the canopy structure, geometric complexities of hundreds of gaps and slits, and the contact between the parachutes of the cluster. In FSI computation of parachutes with such "modified geometric porosity," the flow through the "windows" created by the removal of the panels and the wider gaps created by the removal of the sails cannot be accurately modeled with the Homogenized Modeling of Geometric Porosity (HMGP), which was introduced to deal with the hundreds of gaps and slits. The flow needs to be actually resolved. All these computational challenges need to be addressed simultaneously in FSI modeling of clusters of spacecraft parachutes with modified geometric porosity. The core numerical technology is the Stabilized Space–Time FSI (SSTFSI) technique, and the contact between the parachutes is handled with the Surface-Edge-Node Contact Tracking (SENCT) technique. In the computations reported here, in addition to the SSTFSI and SENCT techniques and HMGP, we use the special techniques we have developed for removing

the numerical spinning component of the parachute motion and for restoring the mesh integrity without a remesh. We present results for 2- and 3-parachute clusters with two different payload models. We also present the FSI computations we carried out for a single, subscale modified-porosity parachute.

Acknowledgments

It doesn't seem like long ago that I was nearing high school graduation and in a frantic scramble to figure out what I wanted to do with my life. The journey that I have been on since that moment is beyond anything that I could have imagined. Countless people have come into my life, many with impeccable timing, to celebrate my highs, help me through the rough times, and guide me in moving forward in my life and career. My time at Rice is no exception, and there are several people to whom I am greatly indebted and very grateful.

Dr. Tayfun Tezduyar provided me with the opportunity to pursue this graduate degree at Rice University. It has been one of the most memorable opportunities in my life, and the lessons I've learned, knowledge I've acquired, and memories I've made will continue to help me in my personal life and career. I'm very thankful for the trust he put in me to uphold the reputation of his research group.

Dr. Kenji Takizawa is the hardest working person I have ever met in my life. His creativity and commitment to his students and research are inspiring, and the many publications and awards he's received in his relatively short career back that up. Without his expertise and words of wisdom, the research presented in this thesis would not have been remotely possible. I am very grateful for his guidance and blatant disregard for anything resembling a normal sleep schedule.

One of the greatest aspects of the T★AFSM is the team atmosphere, and I would not have completed this degree without the help of each one of my teammates. Darren

Montes, Kat Schjodt, and Tony Puntel were extremely patient and knowledgeable in training me. Nik Kostov's guidance was a constant blessing, especially once the aforementioned teammates graduated. His constant words of wisdom and motivation were always there to push me through to the end. Spenser, thanks for being a great roommate and teammate. I'm looking forward to another great couple years in pilot training. To the rest of the team, Cody Boswell, Austin Buscher, Casey Habluetzel, and Ryan Kolesar, thanks for being great teammates. I couldn't have completed this thesis without your help.

Finally, I owe a big thanks to my parents and my brother Evan for putting up with me pursuing such crazy dreams so far away from home. I'm so thankful to have such a wonderful and supportive family. I wouldn't be anything without having you all behind me.

This work was supported in part by NASA Johnson Space Center grant NNX13AD87G. It was also supported by the Rice–Waseda Research Agreement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Manned space flight has been a dream of humans for millenia. It is only within the past century that technology has evolved enough to make travelling to space and returning intact a reality, and within that relatively short time frame, the methods and means of transport have evolved. A complete engineering analysis must be performed for each new spacecraft that is designed, and advanced computational methods and supercomputing resources make this daunting task a little more manageable. The reentry system is arguably the most important aspect of spacecraft design, as a mistake can turn an otherwise successful mission into a disaster. For spacecraft that incorporate parachutes into their reentry system, selecting the perfect parachute design can be very difficult. This thesis shows how continually evolving computational methods can be a valuable resource in the efficient and correct design of the parachute system for spacecraft. The rest of the introductory material in this section is from [59].

Fluid–structure interaction (FSI) modeling of ringsail spacecraft parachute clusters poses a number of computational challenges [57, 13]. These include the lightness of the parachute canopy compared to the air masses involved in the parachute dynamics, the geometric porosity created by the construction of the canopy from "rings" and "sails" with hundreds of "ring gaps" and "sail slits," and the contact between the

parachutes of the cluster. The Team for Advanced Flow Simulation and Modeling (T★AFSM) has been addressing these computational challenges with the Stabilized Space–Time FSI (SSTFSI) technique [75], which was developed and improved over the years by the T★AFSM and serves as the core numerical technology, and a number of special techniques developed in conjunction with the SSTFSI technique.

The SSTFSI technique originates from the Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [63, 69, 71, 65] and its new versions [75, 55, 58]. The DSD/SST formulation is a general-purpose moving-mesh (interface-tracking) method for flows with moving interfaces. Its stabilization parts are the Streamline-Upwind/Petrov-Galerkin (SUPG) [16] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [63, 72] methods. The DSD/SST method is used with the advanced mesh update methods [70, 61, 27, 64, 75] developed by the T★AFSM. Mesh update includes moving the mesh for as long as possible and remeshing when needed. The ST approach, with higher-order functions in time, gives us more effective ways of mesh moving and remeshing (see [48, 46, 49, 50, 47]). While the Arbitrary Lagrangian–Eulerian (ALE) finite element formulation [26] is the most commonly used moving-mesh approach in FSI computations (see, for example, [34, 84, 4, 31, 15, 3, 22, 5, 7, 17, 11, 10, 6, 8, 24, 33, 9]), the DSD/SST formulation now also has a good record of being applied to some of the most challenging FSI computations (see [75, 82, 57, 43, 58, 56, 13, 12] and references therein).

Parachute FSI computations of the T★AFSM with the DSD/SST method precede the development of the SSTFSI technique and the associated special techniques. These computations started as early as 1997 (see [41]), with 3D computations going as far back as 2000 [28, 37], and with a good number of parachute FSI problems solved (see [62, 38, 42, 40, 39, 77, 80]) before reaching the SSTFSI technique. However, it was the SSTFSI technique, and the special techniques developed in conjunction with it, that brought the parachute FSI computations to a new era in addressing some of

the most formidable computational challenges and truly supporting actual parachute design and testing (see [75, 78, 79, 83, 52, 60, 54, 53, 57, 45, 50, 56]).

In an FSI computation with a moving-mesh method, the FSI coupling technique determines how the coupling between the equation blocks representing the fluid mechanics, structural mechanics, and mesh moving equations is handled. The coupling techniques used in the T★AFSM parachute computations evolved from block-iterative FSI coupling [66] (see [77, 75] for the terminology) used in the computations reported in [28, 37, 62, 38] to a more robust version of block-iterative coupling [66, 77, 80, 67] and to quasi-direct coupling [77, 80] and direct coupling [77, 80] techniques. The quasi-direct and direct coupling techniques, which are applicable to cases with nonmatching fluid and structure meshes at the interface, yield more robust algorithms for FSI computations where the structure is light, such as parachute FSI computations. The SSTFSI technique is based on the new versions of the quasi-direct and direct coupling techniques with upgraded and additional interface projection methods [75, 78, 51, 83, 52], has a substantially increased robustness in FSI computations, and rendered the earlier ST FSI solvers obsolete. These new quasi-direct and direct coupling techniques automatically reduce to "monolithic" coupling when the interface has matching fluid and structure meshes. Allowing nonmatching meshes at the interface substantially increases the scope of the FSI solver, leading to success in FSI modeling of challenging problems, such as ringsail spacecraft parachutes (see [78, 79, 83, 52, 60, 54, 53, 57, 45, 50, 56]).

A good number of special FSI techniques were introduced in [75, 78, 51, 83, 52, 60, 54, 53, 45, 50] in conjunction with the SSTFSI technique. These special techniques are mostly in the category of interface projection techniques. They include the FSI Geometric Smoothing Technique (FSI-GST) [75], Separated Stress Projection (SSP) [78, 52], Homogenized Modeling of Geometric Porosity (HMGP) [78], adaptive HMGP [83], "symmetric FSI" method [83], accounting for fluid forces act-

ing on structural components (e.g. parachute suspension lines) that are not expected to influence the flow [83], new versions of the HMGP that are called "HMGP-FG" [52] and "HMGP-FGR" [45], and other interface projection techniques [51]. The special FSI techniques in other categories include the Surface-Edge-Node Contact Tracking (SENCT) technique [75], which is a contact algorithm, multiscale sequentially-coupled FSI techniques [83, 60], rotational-periodicity techniques [52, 60], a new, conservative version of the SENCT technique [54], computed-data reduction techniques [54, 53], intra-canopy versions of the contact algorithm [45], and using higher-order temporal functions in mesh moving [50].

The ringsail spacecraft parachutes the T★AFSM has been focusing on are very large, made of a large number of gores. A gore is the slice of the canopy between two radial reinforcement cables running from the parachute vent to the skirt. The construction of the canopy from rings and sails happens at the gore level. With the HMGP, we bypass the intractable complexities of the geometric porosity by approximating it at the fluid interface with an "equivalent," locally-varying "homogenized" porosity. This is obtained from an HMGP computation with an n-gore slice of the parachute canopy where the flow through the ring gaps and sail slits is actually resolved (see [78, 79, 52, 60, 45] for details). In the earlier HMGP computations with a 4-gore slice, slip conditions were applied on the boundaries intersecting the canopy. With the rotational-periodicity techniques, less constraining conditions can be imposed on those boundaries [52, 60].

Spacecraft parachutes are typically used in clusters of two or three parachutes. The computational challenge associated with the contact between the parachutes of a cluster is addressed with the conservative version of the SENCT technique [54], which is also more robust than the original SENCT technique [75]. During the FSI computation, there might also be a contact within a canopy. This could be a contact between the gores of a parachute canopy, or even a contact between the nodes of a

gore. These computational challenges are addressed with the intra-canopy versions of the contact algorithm [45].

As an additional computational challenge, the ringsail parachute canopy might, by design, have some of its panels and sails removed. The purpose is to increase the aerodynamic performance of the parachute. In FSI computation of parachutes with such "modified geometric porosity," the flow through the "windows" created by the removal of the panels and the wider gaps created by the removal of the sails cannot be accurately modeled with the HMGP and needs to be actually resolved. This and the other computational challenges described in the earlier paragraphs all need to be addressed simultaneously in FSI modeling of clusters of spacecraft parachutes with modified geometric porosity. This is what we have succeeded in doing to a large extent for the computations reported in this thesis.

In the parachute FSI computations reported here, we also use two special techniques that we have developed. With one of the special techniques we remove the numerical spinning component of the parachute motion. We believe that this non-physical spinning has its roots in two of our modeling features: a) the parachute risers are modeled with cable structures, which are currently unable to represent the torsional stiffness that the risers should have, b) in the HMGP, the fluid interface does not see the "valleys" and "peaks" of the gores that the structure interface has, currently not even at some milder level, offering basically no aerodynamic resistance to spinning. Therefore, once the spinning is triggered, for example by an unsymmetric vortex, there is essentially no resistance to it. With the other one of the special techniques, we restore the mesh integrity lost during the mesh motion without resorting to remeshing. The loss of mesh integrity, though not frequent because of the advanced mesh moving methods we are using, should be expected in FSI computations with the level of complexity we have in clusters of ringsail parachutes with modified geometric porosity. When we face such a loss of mesh integrity, as an alternative to remeshing,

we use a technique where the mesh is "relaxed" without altering the mesh at the fluid–structure interface and thus the mesh integrity is restored to some extent. This is of course a less costly and less disruptive alternative to remeshing.

The governing equations are given in Chapter 2. The two special techniques are described in Chapter 3. The computational conditions and results for 2- and 3-parachute clusters with two different payload models are presented in Chapter 4, together with the computations for a single subscale modified-porosity parachute. The concluding remarks are given in Chapter 5.

Chapter 2

Governing Equations and Core FSI Methods

The material in this chapter is from [50].

2.1 Governing Equations of Fluid Mechanics

Let $\Omega_t \subset \mathbb{R}^{n_{\text{sd}}}$ be the spatial domain with boundary Γ_t at time $t \in (0, T)$. The subscript t indicates the time-dependence of the domain. The Navier–Stokes equations of incompressible flows are written on Ω_t and $\forall t \in (0, T)$ as

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} - \mathbf{f} \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \mathbf{0}, \tag{2.1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0, \tag{2.2}$$

where ρ, $\mathbf{u}$ and $\mathbf{f}$ are the density, velocity and the external force, respectively. The stress tensor $\boldsymbol{\sigma}$ is defined as $\boldsymbol{\sigma}(p, \mathbf{u}) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$, with $\boldsymbol{\varepsilon}(\mathbf{u}) = \left((\boldsymbol{\nabla}\mathbf{u}) + (\boldsymbol{\nabla}\mathbf{u})^T \right)/2$. Here p is the pressure, $\mathbf{I}$ is the identity tensor, $\mu = \rho\nu$ is the viscosity, ν is the kinematic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor. The essential and natural boundary conditions for Eq. (2.1) are represented as $\mathbf{u} = \mathbf{g}$ on $(\Gamma_t)_{\text{g}}$ and $\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h}$ on

$(\Gamma_t)_{\mathrm{h}}$, where $(\Gamma_t)_{\mathrm{g}}$ and $(\Gamma_t)_{\mathrm{h}}$ are complementary subsets of the boundary Γ_t, $\mathbf{n}$ is the unit normal vector, and $\mathbf{g}$ and $\mathbf{h}$ are given functions. A divergence-free velocity field $\mathbf{u}_0(\mathbf{x})$ is specified as the initial condition.

2.2 Governing Equations of Structural Mechanics

Let $\Omega_t^s \subset \mathbb{R}^{n_{\mathrm{xd}}}$ be the spatial domain with boundary Γ_t^s, where $n_{\mathrm{xd}} = 3$ for the continuum element, $n_{\mathrm{xd}} = 2$ for membranes, and $n_{\mathrm{xd}} = 1$ for cables. The superscript "s" indicates the structure. The parts of Γ_t^s corresponding to the essential and natural boundary conditions are represented by $(\Gamma_t^s)_{\mathrm{g}}$ and $(\Gamma_t^s)_{\mathrm{h}}$. The equations of motion are written as

$$\rho^s \left(\frac{d^2\mathbf{y}}{dt^2} + \eta \frac{d\mathbf{y}}{dt} - \mathbf{f}^s \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}^s = \mathbf{0}, \tag{2.3}$$

where ρ^s, $\mathbf{y}$, η, $\mathbf{f}^s$ and $\boldsymbol{\sigma}^s$ are the material density, structural displacement, damping coefficient, external force and the Cauchy stress tensor, respectively. The stresses are expressed in terms of the second Piola–Kirchoff stress tensor $\mathbf{S}$, which is related to the Cauchy stress tensor through a kinematic transformation. For the classes of FSI problems the T★AFSM has been focusing on, what makes one structural element model different from the other is the manner in which $\mathbf{S}$ is defined. These definitions can be found in earlier T★AFSM publications[75, 76, 57].

2.3 Core FSI Methods

2.3.1 DSD/SST Formulation of Fluid Mechanics

A space–time variational formulation of incompressible flows (see for example [63, 69, 71, 65, 75, 55]) is written over a sequence of N space–time slabs Q_n, where Q_n is the slice of the space–time domain between the time levels t_n and t_{n+1}, and P_n is the lateral boundary of Q_n. We denote the trial and test functions spaces for the velocity and pressure as $\mathbf{u} \in \mathcal{S}_{\mathbf{u}}$, $p \in \mathcal{S}_p$, $\mathbf{w} \in \mathcal{V}_{\mathbf{u}}$ and $q \in \mathcal{V}_p$. The notation $(\cdot)_n^-$ and $(\cdot)_n^+$ denotes the function values at t_n as approached from below and above. At each time step, the integrations are performed over Q_n. The essential and natural boundary conditions are enforced over $(P_n)_{\mathrm{g}}$ and $(P_n)_{\mathrm{h}}$, the complementary subsets of the lateral boundary of the space–time slab. In the DSD/SST method [63, 69, 71, 65, 75, 55], the space–time finite element interpolation functions are continuous within a space–time slab, but discontinuous from one space–time slab to another. Each Q_n is decomposed into elements Q_n^e, where $e = 1, 2, \ldots, (n_{\mathrm{el}})_n$. The subscript n used with n_{el} is for the general case where the number of space–time elements may change from one space–time slab to another. The finite-dimensional trial and test functions spaces are denoted as $\left(\mathcal{S}_{\mathbf{u}}^h\right)_n$, $\left(\mathcal{S}_p^h\right)_n$, $\left(\mathcal{V}_{\mathbf{u}}^h\right)_n$ and $\left(\mathcal{V}_p^h\right)_n$.

The DSD/SST formulation (from [65]) is written as follows: given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in \left(\mathcal{S}_{\mathbf{u}}^h\right)_n$ and $p^h \in \left(\mathcal{S}_p^h\right)_n$, such that $\forall \mathbf{w}^h \in \left(\mathcal{V}_{\mathbf{u}}^h\right)_n$ and $\forall q^h \in \left(\mathcal{V}_p^h\right)_n$:

$$
\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \; dQ \\
&- \int_{(P_n)_{\mathrm{h}}} \mathbf{w}^h \cdot \mathbf{h}^h \; dP + \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \; dQ + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) d\Omega \\
&+ \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \frac{1}{\rho} [\tau_{\mathrm{SUPG}} \rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) + \tau_{\mathrm{PSPG}} \boldsymbol{\nabla} q^h] \cdot [\mathrm{L}(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h] dQ \\
&+ \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \nu_{\mathrm{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho \boldsymbol{\nabla} \cdot \mathbf{u}^h \; dQ = 0,
\end{aligned}
\tag{2.4}
$$

where

$$Ł(q^h, \mathbf{w}^h) = \rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(q^h, \mathbf{w}^h). \tag{2.5}$$

This formulation is applied to all space–time slabs Q_0, Q_1, Q_2, …, Q_{N-1}, starting with $(\mathbf{u}^h)_0^- = \mathbf{u}_0$. Here τ_{SUPG}, τ_{PSPG} and ν_{LSIC} are the SUPG, PSPG and LSIC stabilization parameters. There are various ways of defining these parameters. Here we provide the definitions given in [65]:

$$\tau_{\text{SUPG}} = \left(\frac{1}{\tau_{\text{SUGN12}}^2} + \frac{1}{\tau_{\text{SUGN3}}^2} \right)^{-\frac{1}{2}}, \tag{2.6}$$

$$\tau_{\text{SUGN12}} = \left(\sum_{a=1}^{n_{\text{en}}} \left| \frac{\partial N_a}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} N_a \right| \right)^{-1}, \tag{2.7}$$

$$\tau_{\text{SUGN3}} = \frac{h_{\text{RGN}}^2}{4\nu}, \tag{2.8}$$

$$h_{\text{RGN}} = 2 \left(\sum_{a=1}^{n_{\text{en}}} |\mathbf{r} \cdot \boldsymbol{\nabla} N_a| \right)^{-1}, \qquad \mathbf{r} = \frac{\boldsymbol{\nabla} \|\mathbf{u}^h\|}{\| \boldsymbol{\nabla} \|\mathbf{u}^h\| \|}, \tag{2.9}$$

$$\tau_{\text{PSPG}} = \tau_{\text{SUPG}}, \tag{2.10}$$

and in [75]:

$$\nu_{\text{LSIC}} = \tau_{\text{SUPG}} \, \|\mathbf{u}^h - \mathbf{v}^h\|^2, \tag{2.11}$$

where n_{en} is the number of (space–time) element nodes and N_a is the space–time shape function associated with the space–time node a. As an alternative to the construction of τ_{SUPG} as given by Eqs. (2.6)–(2.7), another option was introduced in [75]. In that option, τ_{SUPG} is constructed based on separate definitions for the

advection-dominated and transient-dominated limits:

$$\tau_{\text{SUPG}} = \left(\frac{1}{\tau_{\text{SUGN1}}^2} + \frac{1}{\tau_{\text{SUGN2}}^2} + \frac{1}{\tau_{\text{SUGN3}}^2} \right)^{-\frac{1}{2}}, \tag{2.12}$$

$$\tau_{\text{SUGN1}} = \left(\sum_{a=1}^{n_{\text{en}}} \left| \left(\mathbf{u}^h - \mathbf{v}^h \right) \cdot \boldsymbol{\nabla} N_a \right| \right)^{-1}, \tag{2.13}$$

$$\tau_{\text{SUGN2}} = \frac{\Delta t}{2}, \tag{2.14}$$

where $\mathbf{v}^h$ is the mesh velocity and Δt is the time-step size. It was noted in [75] that separating τ_{SUGN12} into its advection- and transient-dominated components as given by Eqs. (2.13)–(2.14) is equivalent to excluding the $\frac{\partial N_a}{\partial t}\big|_{\boldsymbol{\xi}}$ part of $\frac{\partial N_a}{\partial t}$ in Eq. (2.7), making that the definition for τ_{SUGN1}, and accounting for $\frac{\partial N_a}{\partial t}\big|_{\boldsymbol{\xi}}$ in the definition for τ_{SUGN2} given by Eq. (2.14). Here $\boldsymbol{\xi}$ is the vector of element coordinates. For more ways of calculating τ_{SUPG}, τ_{PSPG} and ν_{LSIC}, see [73, 65, 1, 2, 81, 74, 19, 68, 35, 18, 25, 20, 21].

Remark 1 *As an alternative to how the SUPG test function is defined in Eq. (2.4), another option was proposed in [75], where the SUPG test function* $\left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right)$ *is replaced with* $\left(\left(\mathbf{u}^h - \mathbf{v}^h \right) \cdot \boldsymbol{\nabla} \mathbf{w}^h \right)$. *This replacement is equivalent to excluding the* $\frac{\partial \mathbf{w}^h}{\partial t}\Big|_{\boldsymbol{\xi}}$ *part of* $\frac{\partial \mathbf{w}^h}{\partial t}$. *In [75], this option was called "WTSE," and the option where the* $\frac{\partial \mathbf{w}^h}{\partial t}\Big|_{\boldsymbol{\xi}}$ *term is active, "WTSA."*

Remark 2 *With the function spaces defined in the paragraph preceding Eq. (2.4), for each space–time slab velocity and pressure assume double unknown values at each spatial node. One value corresponds to the lower end of the slab, and the other one the upper end. In [75], the option of using double unknown values at a spatial node is called "DV" for velocity and "DP" for pressure. In this case, as pointed out in [75], we use two integration points over the time interval of the space–time slab, and this time-itegration option is called "TIP2." This version of the DSD/SST formulation, with the options set DV, DP and TIP2, is called "DSD/SST-DP."*

Remark 3 *In [75], the option of using, for each space–time slab, a single unknown pressure value at each spatial node was proposed with the option name "SP." With this, another version of the DSD/SST formulation was proposed in [75], where the options set is DV, SP and TIP2. This version is called "DSD/SST-SP." Because the number of pressure unknowns is halved, the computational cost is reduced.*

Remark 4 *To reduce the computational cost further, the option of using only one integration point over the time interval of the space–time slab was proposed in [75]. This time-itegration option is called "TIP1." With that, a third version of the DSD/SST formulation was proposed in [75], where the options set is DV, SP and TIP1. This version is called "DSD/SST-TIP1."*

Remark 5 *As a third way of reducing the computational cost, the option of using, for each space–time slab, a single unknown velocity value at each spatial node was proposed in [75] with the option name "SV." In the SV option, of the two parts of Eq. (2.4), the one generated by* $(\mathbf{w}^h)_n^+$ *is removed, and we explicitly set* $(\mathbf{u}^h)_n^+ = (\mathbf{u}^h)_n^-$*, which makes the velocity field continuous in time. Based on the SV option, a fourth version of the DSD/SST formulation was proposed in [75], where the options set is SV, SP and TIP1. This version is called "DSD/SST-SV." In this version of the DSD/SST formulation, as it was proposed in [75], one can use the SUPG test function option WTSE.*

Remark 6 *For DSD/SST-SP, DSD/SST-TIP1 and DSD/SST-SV, in integration of the incompressibility-constraint term over each space–time slab, as proposed in [44], we use only one integration point in time, shifted to the upper time level of the slab. All other terms in the space–time finite element formulation are integrated by using Gaussian quadrature points in time, with the number of points set to whatever we intended to have for the overall formulation. With this technique, as pointed in [44], the incompressibility constraint equation focuses on the velocity field* $(\mathbf{u}^h)_{n+1}^-$.

2.3.2 Semi-Discrete Formulation of Structural Mechanics

With $\mathbf{y}^h$ and $\mathbf{w}^h$ coming from appropriately defined trial and test function spaces, respectively, the semi-discrete finite element formulation of the structural mechanics equations (see [30, 14, 37]) is written as

$$\int_{\Omega_0^s} \mathbf{w}^h \cdot \rho^s \frac{d^2\mathbf{y}^h}{dt^2}\, d\Omega + \int_{\Omega_0^s} \mathbf{w}^h \cdot \eta\rho^s \frac{d\mathbf{y}^h}{dt}\, d\Omega + \int_{\Omega_0^s} \delta\mathbf{E}^h : \mathbf{S}\, d\Omega = \int_{\Omega_t^s} \mathbf{w}^h \cdot \left(\mathbf{t}^h + \rho^s \mathbf{f}^s\right) d\Omega. \tag{2.15}$$

The fluid mechanics forces acting on the structure are represented by vector $\mathbf{t}^h$. The above formulation is for structures represented by a membrane model. The left-hand-side terms of Eq. (2.15) are referred to in the original configuration and the right-hand-side terms in the deformed configuration at time t. Time discretization of Eq. (2.15) is based on the Hilber–Hughes–Taylor scheme [23].

Remark 7 *In the computations reported here and those reported earlier by the T★AFSM, the mass matrix associated with the first term of Eq. (2.15) is lumped.*

2.3.3 Stabilized Space–Time Fluid–Structure Interaction technique

The SSTFSI technique was introduced in [75], where it was described based on the finite element formulations given by Eqs. (2.4) and (2.15), with a slight change of notation and with a clarification of how the fluid–structure interface conditions are handled. In that notation subscripts 1 and 2 refer to fluid and structure, respectively. Furthermore, while subscript I refers to the fluid–structure interface, subscript E refers to "elsewhere" in the fluid and structure domains or boundaries. Here we write

from [75] the equations representing the SSTFSI technique:

$$\int_{Q_n} \mathbf{w}_{1\mathrm{E}}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}_{1\mathrm{E}}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \; dQ$$
$$- \int_{(P_n)_{\mathrm{h}}} \mathbf{w}_{1\mathrm{E}}^h \cdot \mathbf{h}_{1\mathrm{E}}^h \; dP + \int_{Q_n} q_{1\mathrm{E}}^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \; dQ + \int_{\Omega_n} (\mathbf{w}_{1\mathrm{E}}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) d\Omega$$
$$+ \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \frac{1}{\rho} [\tau_{\mathrm{SUPG}} \rho \left(\frac{\partial \mathbf{w}_{1\mathrm{E}}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}_{1\mathrm{E}}^h \right) + \tau_{\mathrm{PSPG}} \boldsymbol{\nabla} q_{1\mathrm{E}}^h] \cdot [\mathrm{L}(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h] dQ$$
$$+ \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \nu_{\mathrm{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}_{1\mathrm{E}}^h \rho \boldsymbol{\nabla} \cdot \mathbf{u}^h \; dQ = 0, \tag{2.16}$$

$$\int_{Q_n} q_{1\mathrm{I}}^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \; dQ + \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \frac{1}{\rho} [\tau_{\mathrm{PSPG}} \boldsymbol{\nabla} q_{1\mathrm{I}}^h] \cdot [\mathrm{L}(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h] dQ = 0, \tag{2.17}$$

$$\int_{(\Gamma_{1\mathrm{I}})_{\mathrm{REF}}} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1} \cdot \left((\mathbf{x}_{1\mathrm{I}}^h)_{n+1} - (\mathbf{x}_{2\mathrm{I}}^h)_{n+1} \right) d\Gamma = 0, \tag{2.18}$$

$$\int_{(\Gamma_{1\mathrm{I}})_{\mathrm{REF}}} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \cdot \left((\mathbf{u}_{1\mathrm{I}}^h)_{n+1}^- - \mathbf{u}_{2\mathrm{I}}^h \right) d\Gamma = 0, \tag{2.19}$$

$$\int_{(P_n)_{\mathrm{h}}} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \cdot \mathbf{h}_{1\mathrm{I}}^h \; dP = - \int_{(P_n)_{\mathrm{h}}} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \cdot p^h \mathbf{n} \; dP$$
$$+ \int_{Q_n} 2\mu \boldsymbol{\varepsilon}((\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^-) : \boldsymbol{\varepsilon}(\mathbf{u}^h) \; dQ + \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \cdot \boldsymbol{\nabla} \cdot \left(2\mu \boldsymbol{\varepsilon}(\mathbf{u}^h) \right) \; dQ, \tag{2.20}$$

$$\int_{(\Omega_{2\mathrm{I}})_{\mathrm{REF}}} \mathbf{w}_{2\mathrm{I}}^h \cdot \left(\mathbf{h}_{2\mathrm{I}}^h + \left(\mathbf{h}_{1\mathrm{I}}^h \right)_{\mathrm{A}} + \left(\mathbf{h}_{1\mathrm{I}}^h \right)_{\mathrm{B}} \right) d\Omega = 0, \tag{2.21}$$

$$\int_{(\Omega_2)_0} \mathbf{w}_2^h \cdot \rho_2 \frac{d^2\mathbf{y}^h}{dt^2}\ d\Omega + \int_{(\Omega_2)_0} \mathbf{w}_2^h \cdot \eta\rho_2 \frac{d\mathbf{y}^h}{dt}\ d\Omega + \int_{(\Omega_2)_0} \delta\mathbf{E}^h : \mathbf{S}^h\ d\Omega =$$
$$\int_{\Omega_2} \mathbf{w}_2^h \cdot \rho_2 \mathbf{f}_2^h\ d\Omega + \int_{\Omega_{2\mathrm{E}}} \mathbf{w}_{2\mathrm{E}}^h \cdot \mathbf{h}_{2\mathrm{E}}^h\ d\Omega + \int_{\Omega_{2\mathrm{I}}} \mathbf{w}_{2\mathrm{I}}^h \cdot \mathbf{h}_{2\mathrm{I}}^h\ d\Omega. \tag{2.22}$$

Here $(\Gamma_{1\mathrm{I}})_{\mathrm{REF}}$ and $(\Omega_{2\mathrm{I}})_{\mathrm{REF}}$ represent some reference configurations of $\Gamma_{1\mathrm{I}}$ and $\Omega_{2\mathrm{I}}$, and $\mathbf{x}_{1\mathrm{I}}^h$ and $\mathbf{x}_{2\mathrm{I}}^h$ are the fluid mechanics and structural mechanics nodal positions at the fluid–structure interface. In reconciling the slightly modified notation used here with the notation used in Eqs. (2.4) and (2.15), we note that $\rho_2 = \rho^s$, $\mathbf{f}_2^h = \mathbf{f}^s$, $(\Omega_2)_0 = \Omega_0^s$, $\Omega_2 = \Omega_t^s$, and $\Omega_{2\mathrm{I}}$ and $\Omega_{2\mathrm{E}}$ denote the partitions of Ω_2 corresponding to the interface and elsewhere. We also note that $\mathbf{h}_{2\mathrm{I}}^h = \mathbf{t}^h$, and $\left(\mathbf{h}_{1\mathrm{I}}^h\right)_{\mathrm{A}}$ and $\left(\mathbf{h}_{1\mathrm{I}}^h\right)_{\mathrm{B}}$ represent the values of $\mathbf{h}_{1\mathrm{I}}^h$ associated with the fluid surfaces above and below the membrane structure. The symbol $\mathbf{h}_{2\mathrm{E}}^h$ denotes the prescribed external forces acting on the structure in $\Omega_{2\mathrm{E}}$, which is separate from $\mathbf{f}_2^h$. In this formulation, $(\mathbf{u}_{1\mathrm{I}}^h)_{n+1}^-$, $\mathbf{h}_{1\mathrm{I}}^h$ and $\mathbf{h}_{2\mathrm{I}}^h$ (the fluid velocity, fluid stress and structural stress at the fluid–structure interface) are treated as separate unknowns, and Eqs. (2.19), (2.20) and (2.21) can be seen as corresponding to these three unknowns, respectively. The structural displacement rate at the interface, $\mathbf{u}_{2\mathrm{I}}^h$, is derived from $\mathbf{y}^h$.

Remark 8 *Equation (2.18) represents a projection at the fluid–structure interface that was not explicitly mentioned in [75] but was part of the original SSTFSI technique. It is the projection of the structure nodal positions to the fluid nodal positions. This projection equation is solved at every block-iteration between the fluid+structure block and mesh-moving block (see Section 5 of [75] for the terminology). It is solved with GMRES [36] iterations.*

We note that Eq. (2.20) has been derived by assuming that the viscous-flux jump terms across inter-element borders are negligible. We also note that the last term of that equation, in its original form in [75], was written as a global integral $\int_{Q_n}$ rather than a series of element-level integrals. Alternatively, one can leave that projection

equation in its form prior to the integration-by-parts:

$$\int_{(P_n)_{\mathrm{h}}} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \cdot \mathbf{h}^h_{1\mathrm{I}} \; dP = - \int_{(P_n)_{\mathrm{h}}} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \cdot p^h \mathbf{n} \; dP$$
$$+ \quad \int_{(P_n)_{\mathrm{h}}} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \cdot \left(\mathbf{n} \cdot \left(2\mu \boldsymbol{\varepsilon}(\mathbf{u}^h) \right) \right) \; dP. \tag{2.23}$$

This requires also the projection of $\boldsymbol{\varepsilon}(\mathbf{u}^h)$ from the element interiors to the nodes.

The formulation given by Eqs. (2.16)–(2.22) is based on allowing for cases when the fluid and structure interface meshes are not identical. If they are identical, as pointed out in [75], the same formulation can still be used.

It was noted in [75] that, for constant viscosity, the term $\boldsymbol{\nabla} \cdot \left(2\mu \boldsymbol{\varepsilon}(\mathbf{u}^h) \right)$ in Eq. (2.20) vanishes for tetrahedral elements and in most cases can be neglected for hexahedral elements. It was also noted in [75] that the same can be said about that term as part of $Ł(p^h, \mathbf{u}^h)$ in Eqs. (2.16) and (2.17).

Remark 9 *Although this detail was not discussed in [75], the two projection equations given by Eqs. (2.19) and (2.21) are solved by "numerical substitution," which essentially consists of sub-level GMRES iterations.*

Remark 10 *If the fluid and structure meshes at the fluid–structure interface are identical, then the projections given by Eqs. (2.18), (2.19) and (2.21) simplify to "direct substitution."*

Remark 11 *In [75], the versions of the SSTFSI technique corresponding to the DSD/SST-DP, DSD/SST-SP, DSD/SST-TIP1 and DSD/SST-SV formulations (see Remarks 2–5) were called "SSTFSI-DP," "SSTFSI-SP," "SSTFSI-TIP1" and "SSTFSI-SV," respectively.*

In computations where we account for the porosity of the membrane fabric, as formulated in [75], Eq. (2.19) is replaced with

$$\int_{\Gamma_{1\mathrm{I}}} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \cdot \left((\mathbf{u}^h_{1\mathrm{I}})^-_{n+1} - \mathbf{u}^h_{2\mathrm{I}} + k_{\mathrm{PORO}} \left(\mathbf{n} \cdot \mathbf{h}^h_{1\mathrm{I}} \right) \mathbf{n} \right) d\Gamma = 0, \tag{2.24}$$

where k_{PORO} is the porosity coefficient. In our current implementation, in Eq. (2.24) we take into account only the pressure component of $\mathbf{h}^h_{1\mathrm{I}}$.

Remark 12 *In FSI computations with membranes and shells, the pressure at the interface has split nodal values corresponding to the fluid surfaces above and below the membrane or shell structure. It was proposed in [75] to use such split nodal values for pressure also at the boundaries (i.e. edges) of a membrane structure submerged in the fluid. As pointed out in [75], our computations show that this provides additional numerical stability for the edges of the membrane.*

Chapter 3

Special Techniques

The material in this chapter is from [59].

3.1 Removing Spinning

We first define, as described in Section 4.1.2 of [54], the individual parachute axes:

$$\mathbf{r}_k \equiv (\mathbf{x}_\mathrm{c})_k - \mathbf{x}_\mathrm{p}, \tag{3.1}$$

where k indicates the k^{th} parachute, and $(\mathbf{x}_\mathrm{c})_k$ and $\mathbf{x}_\mathrm{p}$ are the centroid of the parachute and the confluence of the cluster, respectively, and obtain

$$(\mathbf{g}_r)_k = \frac{\mathbf{r}_k}{\|\mathbf{r}_k\|}. \tag{3.2}$$

We consider the angular momentum

$$\mathbf{L}_k = \int_{\Omega_k} \boldsymbol{\zeta} \times \rho \mathbf{u} d\Omega, \tag{3.3}$$

where ρ is the density, $\mathbf{u}$ is the velocity, and $\boldsymbol{\zeta}$ is the vector from the origin. Based

on that, we define an average angular velocity:

$$\overline{\boldsymbol{\omega}}_k = (\mathbf{I}_{\text{total}})_k^{-1}\mathbf{L}_k, \tag{3.4}$$

where $(\mathbf{I}_{\text{total}})_k$ is the inertia tensor. We want to remove the component of that in the $(\mathbf{g}_r)_k$ direction:

$$(\boldsymbol{\omega}_{\text{S}})_k = (\overline{\boldsymbol{\omega}}_k \cdot (\mathbf{g}_r)_k)\,(\mathbf{g}_r)_k. \tag{3.5}$$

That would lead to the following new velocity for the points of the k^{th} parachute:

$$\mathbf{u}_{\text{RS}} = \mathbf{u} - (\boldsymbol{\omega}_{\text{S}})_k \times \boldsymbol{\zeta}. \tag{3.6}$$

Remark 13 *Choosing the origin to be along the parachute axis, which passes through the parachute centroid, makes the velocity modification unique. Here we choose the origin to be the confluence of the cluster.*

3.2 Mesh Relaxation

In the mesh relaxation technique we propose here, the new mesh will have the same number of nodes and elements as before, but certain nodes are moved slightly to improve the quality of certain elements within the domain. For that, we use the large-deformation solid mechanics equations rather than the linear-elasticity equations with Jacobian-based stiffening [70, 61, 27, 64, 75] that we use in mesh moving. One of the advantages of this is the ability to choose from many constitutive models and include geometric stiffness, which means that we can define the undeformed shapes in arbitrary orientations.

To add more flexibility to the equations we are using, we introduce an element-based undeformed shape Ω_0^e for the e^{th} element. This is essentially a shape generated for each element. In general we do not need to relate it to the current domain, but in the mesh relaxation technique we propose here, we do. Given mesh position $\mathbf{X}_{\text{REF}} \in \Omega_{\text{REF}}$ for the current mesh, we solve for the displacement $\mathbf{y}$ and obtain a better quality mesh with $\mathbf{X}_{\text{REF}} + \mathbf{y}$. In solving the solid mechanics equations, we integrate over each element domain Ω_0^e, and the full displacement for an element node a is obtained as follows:

$$\mathbf{x}_a^e = (\mathbf{X}_{\text{REF}}^e)_a - (\mathbf{X}_0^e)_a + \mathbf{y}_a^e. \tag{3.7}$$

By design, the undeformed shape is the shape we want to obtain from solving the solid mechanics equations. We now call those elements "target elements." There are several options for constructing the target element shapes, and the current mesh could be thought of as the starting point.

The first option is setting a minimum desired element volume. If an element in the current mesh has a volume less than that, the volume of the element is increased incrementally until it reaches the minimum desired volume. This option is useful for

increasing the volume of specific elements in a mesh relative to other elements.

The second option is a global volume scaling. It is a factor by which all element volumes are multiplied, thus placing all elements in the mesh in either tension or compression. This is sometimes important for obtaining a stable solution from the solid mechanics equations.

The third option is to shape the element such that it is a regular tetrahedron, has its shape in the current mesh we are relaxing, or is a linear interpolation between the two. The linear interpolation needs to be done carefully. First we define a regular tetrahedron $\mathbf{X}^e_{\mathrm{REG}}$ such that the volume is the same as the volume in the current mesh and the centroid is at the origin. With that, we calculate the following deformation gradient tensor at the element center:

$$\mathbf{F} = \frac{\partial \mathbf{X}^e_{\mathrm{REF}}}{\partial \mathbf{X}^e_{\mathrm{REG}}}, \tag{3.8}$$

and obtain the right Cauchy–Green tensor:

$$\mathbf{C} = \mathbf{F}^T\mathbf{F}. \tag{3.9}$$

After the polar decomposition of $\mathbf{C}$, we obtain the rigid-body rotation tensor $\mathbf{R}$. With this, we exclude the rigid-body rotation for an element node a:

$$\left(\hat{\mathbf{X}}^e_{\mathrm{REG}}\right)_a = \mathbf{R}\left(\mathbf{X}^e_{\mathrm{REG}}\right)_a, \tag{3.10}$$

and obtain the target element shape as

$$\left(\mathbf{X}^e_0\right)_a = (1-s)\left(\mathbf{X}^e_{\mathrm{REF}}\right)_a + s\left(\hat{\mathbf{X}}^e_{\mathrm{REG}}\right)_a. \tag{3.11}$$

Here $0 \leq s \leq 1$, $s = 1$ leads to the target element that is a regular tetrahedron with

the same volume as the volume in the current mesh, and $s = 0$ leads to the element in the current mesh.

The fourth, and the last, option is to specify a minimum aspect ratio to fix: $A_{\min}$. For any element with an aspect ratio less than or equal to that, we use $s = 0$ in the interpolation given by Eq. (3.11). For any element that has an aspect ratio above $A_{\min}$, we use a globally specified s value to relax the shape.

These options, listed in order of precedence, are combined to generate a database of target element shapes. With these element-specific unstressed shapes, the solid mechanics solver attempts to reduce the stress in the new mesh. This makes the elements in the new mesh more closely resemble the target elements, thus increasing the quality of the elements in the mesh.

Relaxing the mesh has several advantages over remeshing. It is much less time consuming, is easier to automate, and reduces the amount of error introduced by projecting data from one mesh to a new mesh.

Chapter 4

Parachute Examples

The material in this chapter is from [59].

4.1 Parachute Clusters

The objective of the parachute cluster computations presented in this thesis is to determine how the newest generation of the NASA Orion Capsule parachute system, the modified-porosity (MP) parachute, performs when used in cluster configurations. The purpose of the MP design is to, compared to previous designs, reduce cluster flyout angles, increase stability, and reduce canopy interactions.

The MP parachute modeled in this thesis has the same physical parameters as the pad abort (PA) parachute modeled in [60, 54], except it has a suspension-line to nominal diameter ratio, L_s/D_o, of 1.44 instead of $L_s/D_o = 1.15$. In addition, every 5th gore on Sail 11 is removed as well as the top 25% of Sail 6. Each parachute in the cluster has 80 gores, a nominal diameter of about 120 ft, 4 rings and 9 sails. The material properties are the same as described in [45].

Four different fully-open MP cluster configurations are modeled, each with a different number of parachutes and payload configuration. A single initial coning angle (θ_{INIT}) is used for each cluster. All 2-parachute MP clusters (2-MP) in this thesis

have $\theta_{\text{INIT}} = 15°$. All 3-parachute MP clusters (3-MP) have $\theta_{\text{INIT}} = 25°$.

The payload is modeled as either a point mass at the confluence of the risers (PAC) or a series of truss elements (PTE) (see [54]). In drop tests, the parachutes are connected to a rectangular pallet that is designed to represent the mass and inertial properties of a proposed crew capsule. The PTE configuration distributes the payload mass at 9 different points to match the mass, center of gravity, and six components of the inertia tensor of the pallet used in drop tests. The points are connected by 5 cable elements and 26 truss elements below the confluence (Figure 4.1). To facilitate

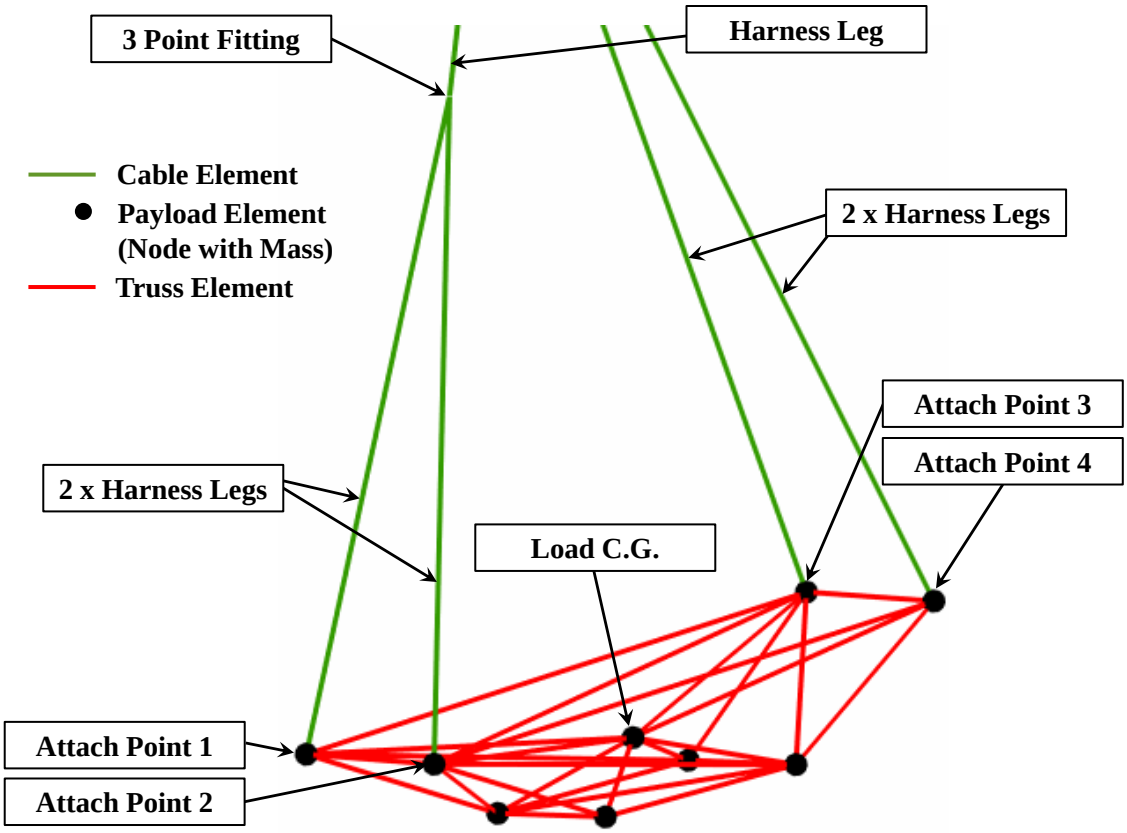


Figure 4.1: Payload as a truss element (PTE) configuration showing the point masses and the cable and truss elements (figure from [32]). The cable elements are the four longer ones and the truss elements are the remaining, shorter ones.

comparison to PA clusters reported in [54], PAC computations use a payload mass of 16,704 lbs, while PTE computations have total payload mass of 19,200 lbs.

Remark 14 *In the computations reported in [54], the payload mass of the PAC configuration was incorrectly reported as 19,200 lbs. The correct weight for this specific configuration is 16,704 lbs, and that is what was used in the computations reported in [54].*

All computations are carried out using air properties at standard sea-level conditions. The density is 2.38×10^{-3} slug/ft^3. The kinematic viscosity is 1.57×10^{-4} ft^2/s.

4.1.1 Starting Conditions

We first build a starting condition for a single parachute. We begin with a parachute shape obtained with the symmetric FSI computation reported in [45]. We do another single-parachute symmetric FSI computation with a horizontal inflow velocity of $U_{\text{ref}}\sin(\theta_{\text{INIT}})$, where U_{ref} is the reference value of the payload descent speed. It is 30 ft/s for 2-MP, and 25.7 ft/s for 3-MP. We compute for three breathing cycles. We use the parachute shape and position corresponding to the time when the parachute skirt diameter is at its average value to assemble the cluster structural mechanics mesh. A cluster of parachutes is generated by duplicating and rotating the structure and interface meshes from the single parachute such that each parachute is at the specified θ_{INIT} and joined together at a confluence.

After that, we generate a fluid mechanics mesh and do a fluid mechanics computation, holding the structural parachute shapes and positions fixed. The inflow velocity is U_{ref}. Next, we do a fluid mechanics computation with a prescribed, time-dependent shape for all parachutes of the cluster. The time-dependent shape comes from the single-parachute symmetric FSI computation carried out earlier with a horizontal inflow velocity of $U_{\text{ref}}\sin(\theta_{\text{INIT}})$. We use the solution from the fluid mechanics computation with prescribed parachute motion as the starting condition for the FSI computation.

4.1.2 Computational Methods and Parameters

The computational domain is cylindrical with a diameter of 1,740 ft and a height of 1,566 ft. Figure 4.2 shows, for a single parachute, the canopy structure mesh and the fluid mechanics interface mesh. The fluid mechanics volume mesh consists of

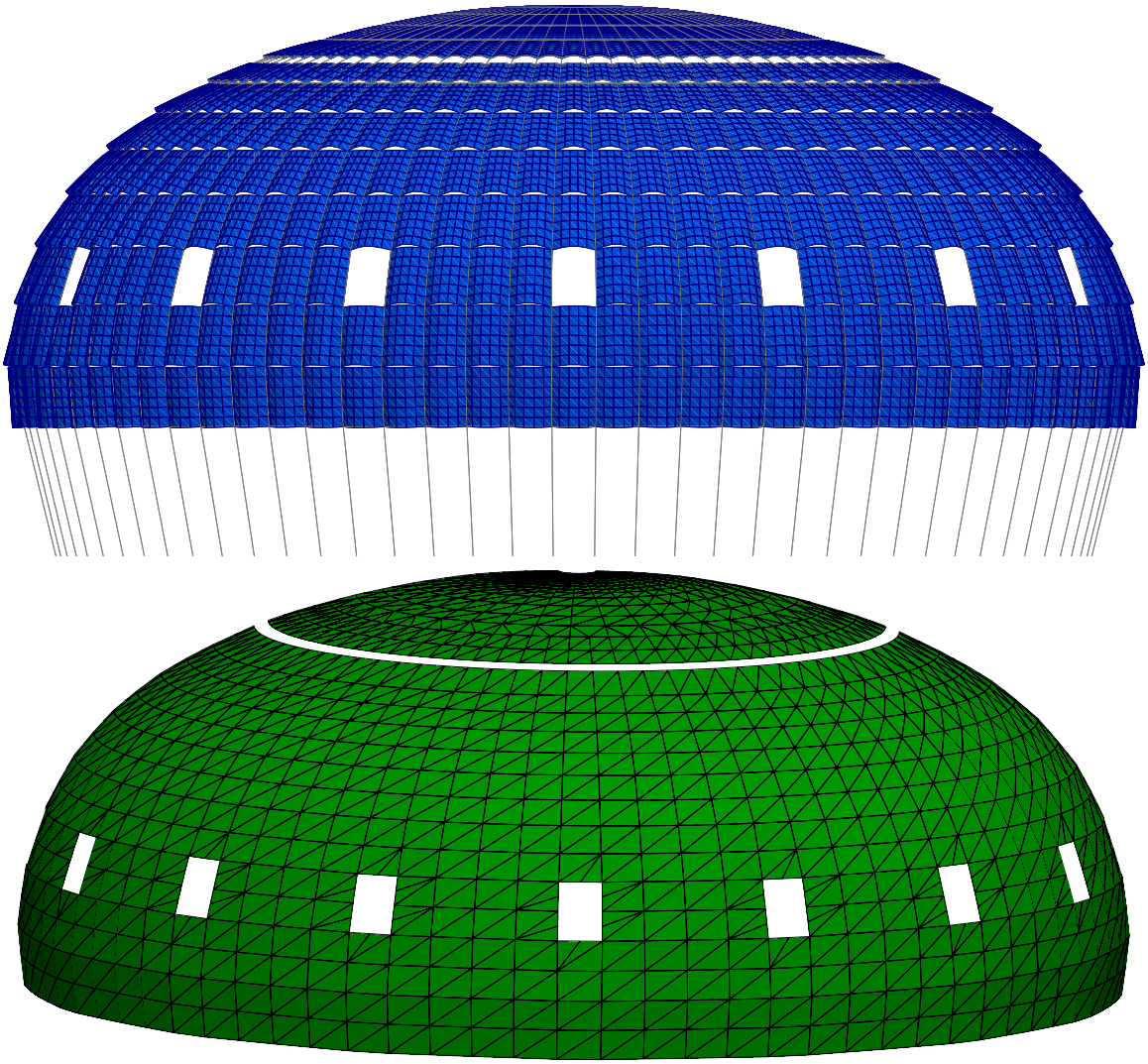


Figure 4.2: Structural mechanics mesh (top) and fluid interface mesh (bottom) for a single MP parachute. For the number of nodes and elements at these interfaces, see Table 4.1.

four-node tetrahedral elements, and the membrane elements used in the parachute structure are quadrilateral. The number of nodes and elements for the fluid and structure are given in Table 4.1. We move the reference frame with a vertical velocity of

Table 4.1: Number of nodes (nn) and elements (ne) for the parachute clusters. The PTE configuration has 10 more structure nodes (8 of them point masses), 5 more cable elements, and 26 truss elements.

		2-MP	3-MP
Structure			
	nn	58,147	87,220
Membrane	ne	48,480	72,720
Cable	ne	24,642	36,963
Payload	ne	1	1
Interface	nn	55,104	82,656
	ne	48,480	72,720
Fluid			
Volume	nn	335,212	307,217
	ne	1,966,851	1,786,212
Interface	nn	4,472	6,708
	ne	8,104	12,156

U_{ref}, and translate the mesh horizontally and vertically with the average displacement rate of the structure beyond the reference velocity U_{ref}. We use the velocity form of the free-stream conditions at the lateral boundaries as well since the mesh translates horizontally.

All computations are carried out in a parallel computing environment. The meshes are partitioned to enhance the parallel efficiency of the computations. Mesh partitioning is based on the METIS [29] algorithm. In solving the linear equation systems encountered at every nonlinear iteration, the GMRES search technique [36] is used with a diagonal preconditioner.

In the symmetric FSI computation with a single MP for three breathing cycles, we use the SSTFSI-TIP1 technique (see Remarks 4 and 11), with the SUPG test function option WTSA (see Remark 1). The stabilization parameters used are those given by Eqs. (2.8)–(2.10), (2.12)–(2.13) and (2.11), with the τ_{SUGN2} term dropped from Eq. (2.12). The porosity model is HMGP-FGR [45]. The values of the HMGP "fabric porosity (k_{F})" and "geometric porosity (k_{G})" for the 14 "patches" we have

(see [78, 52] for the terminology) come essentially from the values used in the PA parachute computations reported in [60, 54, 53]. Those values are given in Table 4.2. The interface projection methods used include the SSP technique [78]. The fully-discretized, coupled fluid and structural mechanics and mesh-moving equations are solved with the quasi-direct coupling technique (see Section 5.2 in [75]). We use selective scaling (see [75]), with the scale for the structure part set to 10 and for the other parts set to 1. The time-step size is 0.0232 s, with 6 nonlinear iterations per time step. The number of GMRES iterations per nonlinear iteration is 90 for the fluid+structure block, and 30 for the mesh-moving block.

Table 4.2: HMGP coefficients k_{F} and k_{G} used in the PA parachute computations reported in [60, 54, 53] (note that Patch 1 and Patch 14 do not have a k_{G} value because those patches do not contain a ring gap or sail slit). We use these values also in the MP parachute computations reported in this thesis, with some special treatment where we created windows and wide gaps. For the patch where we created a wide gap, we use only the corresponding k_{F} value. For the two gore patches that contain a window, we use only the two corresponding k_{F} values. This is what was done also in the single-MP parachute computations reported in [45].

Patch	k_{F} (CFM)	k_{G}
1	125.5	
2	115.0	0.967
3	115.0	0.971
4	75.8	0.960
5	39.9	0.949
6	39.8	0.756
7	39.7	0.769
8	39.7	0.824
9	39.6	0.831
10	68.8	0.816
11	98.3	0.819
12	97.7	0.820
13	97.0	0.867
14	97.1	

The cluster fluid mechanics computations with fixed shapes and positions are done in two parts. The first part uses the semi-discrete formulation given in [65]. We

compute 800 time steps with a time-step size of 0.232 s and 6 nonlinear iterations per time step. The number of GMRES iterations per nonlinear iteration is 90. The second part uses the DSD/SST-TIP1 technique (see Remark 4), with the same SUPG test function option and stabilization parameters as those described above. We compute 1,800 time steps with a time-step size of 0.0232 s, 6 nonlinear iterations per time step, and 90 GMRES iterations per nonlinear iteration. The porosity model is HMGP-FG [52].

For the cluster fluid mechanics computations with prescribed, time-dependent shapes, we use the DSD/SST-TIP1 technique, with the same SUPG test function option and stabilization parameters as those described above. The porosity model is HMGP-FGR. We compute roughly 300 time steps with a time-step size of 0.0232 s and 6 nonlinear iterations per time step. The number of GMRES iterations per nonlinear iteration is 90 for the fluid mechanics block, and 30 for the mesh-moving block.

In the cluster FSI computations we use the SSTFSI-TIP1 technique, with the same SUPG test function option and stabilization parameters as those described above. The porosity model is HMGP-FGR. The interface projection methods used include the SSP technique. The SENCT-FC contact algorithm [54] is used with $\epsilon^{\mathrm{S}} = \epsilon^{\mathrm{C}} = 2.9$ ft. We use the quasi-direct coupling technique and selective scaling, with the scale for the structure part set to 1,000 and for the other parts set to 1. The time-step size is 0.0232 s, and the number of nonlinear iterations per time step is 6. The number of GMRES iterations per nonlinear iteration is 120 for the fluid+structure block, and 30 for the mesh-moving block.

We compute each parachute cluster for about 80 s. The numerical spinning component of the parachute motion is removed using the technique described in Section 3.1, approximately every 150 time steps. The mesh update for the fluid mechanics part includes, as needed, mesh relaxation, as described in Section 3.2, or remeshing. The frequency and the choice between mesh relaxation and remeshing vary for each com-

putation and depend on how much the cluster rotates about the vertical axis and how much each parachute rotates about its own axis.

4.1.3 Results

The critical measure of performance for the parachutes modeled in this thesis is the payload descent speed, not only in terms of its average value, but also in terms of its fluctuations. Drag coefficient is another way of expressing the performance, and is calculated as follows:

$$C_{\mathrm{D}} = \frac{W}{\frac{1}{2}\rho U^2 S_{\mathrm{o}}}, \tag{4.1}$$

where W is the payload weight, ρ is the density of the air, U is the payload descent speed, and S_{o} is the nominal area of the parachute. Figures 4.3 and 4.4 show the payload descent speed and drag coefficient for the MP parachute clusters. Figures 4.5–4.8 show the clusters at different instants. Figures 4.9–4.12 show the vent-separation distances ("L_{VS}") for the clusters. The horizontal black line on each plot shows the approximate vent-separation distance when the parachutes are in contact.

To better understand the descent speed fluctuations, a method to decompose the payload velocity into components based on geometric contributing factors was introduced in [54]. We use that method here to decompose the payload velocity of the MP parachute clusters. Figures 4.13–4.16 show the decomposition results. In those figures, as in [54], $\overline{u}_{\mathrm{B}}$, $\overline{u}_{\mathrm{S}}$, and $\overline{u}_{\mathrm{C}}$ refer to the average, over the number of parachutes, the parachute breathing, swinging, and coning, respectively. The symbols u_{B} and u_{S} represent the breathing and swinging for the cluster, u_{P} is the payload velocity, and $\overline{u}_{\mathrm{A}}$ is the aerodynamic contributor to that. Figures 4.17–4.20 show the individual-parachute contributions to the payload descent speed. To further analyze the results, Figures 4.21–4.24 show the individual-parachute contributions to the drag.

Figures 4.25–4.28 show the payload and canopy-centroid descent speeds.

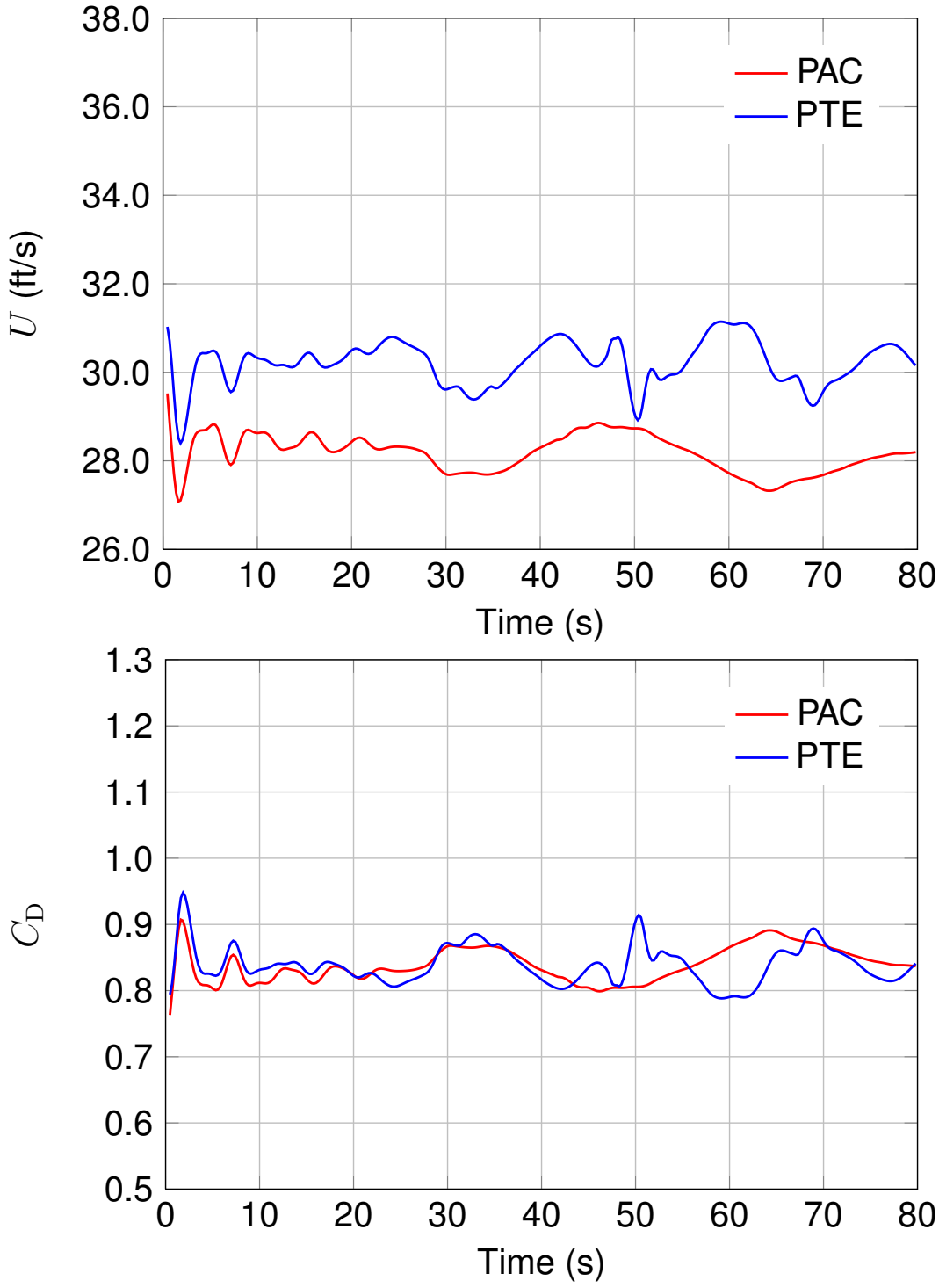


Figure 4.3: Payload descent speed and drag coefficient for 2-MP with PAC and PTE.

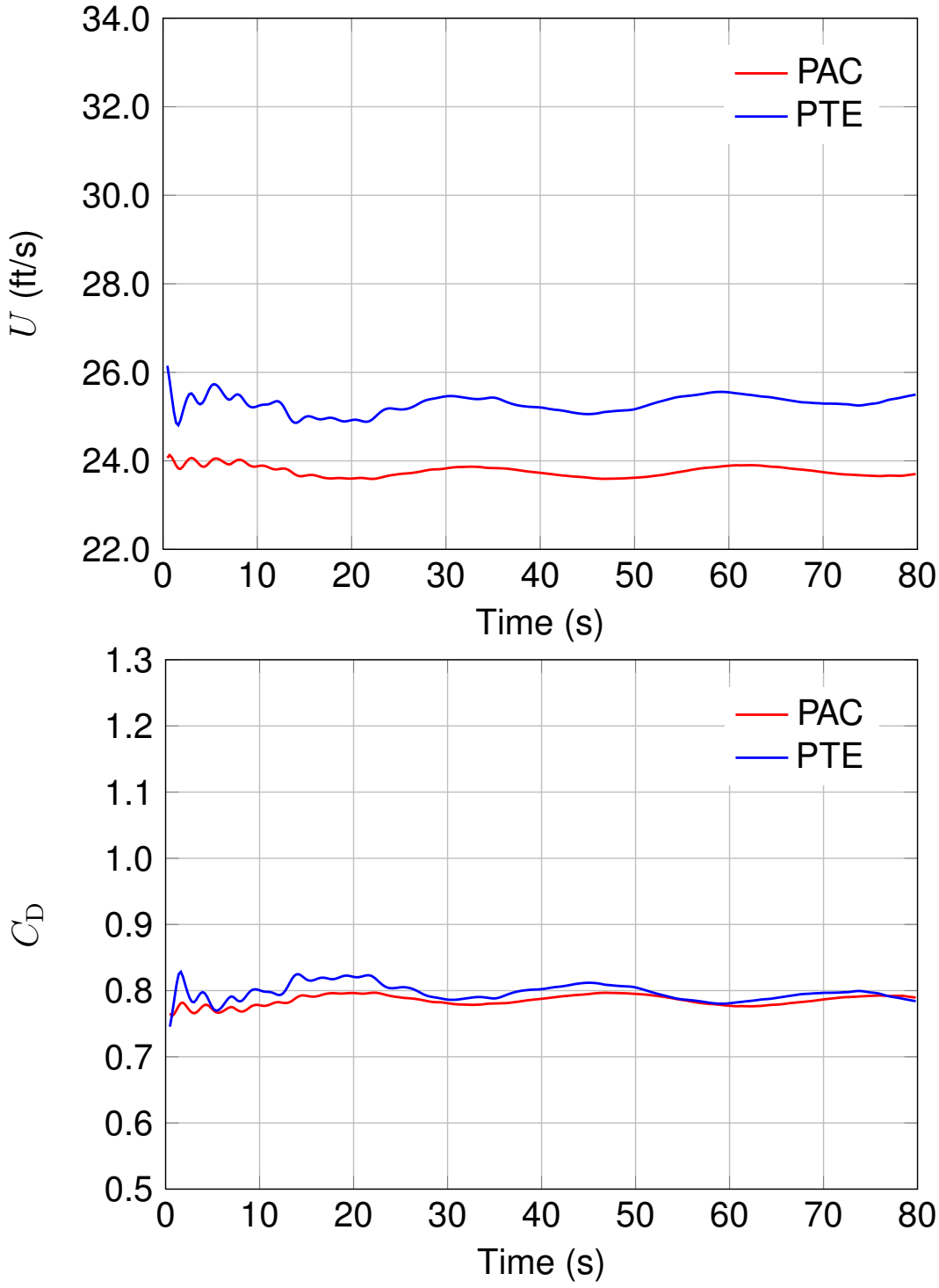


Figure 4.4: Payload descent speed and drag coefficient for 3-MP with PAC and PTE.

Figure 4.5: 2-MP with PTE at $t = 44.08$ s and $t = 46.40$ s.

Figure 4.6: 2-MP with PTE at $t = 47.79$ s and $t = 51.04$ s.

Figure 4.7: 3-MP with PTE at $t = 60.55$ s and $t = 67.28$ s.

Figure 4.8: 3-MP with PTE at $t = 74.47$ s and $t = 79.11$ s.

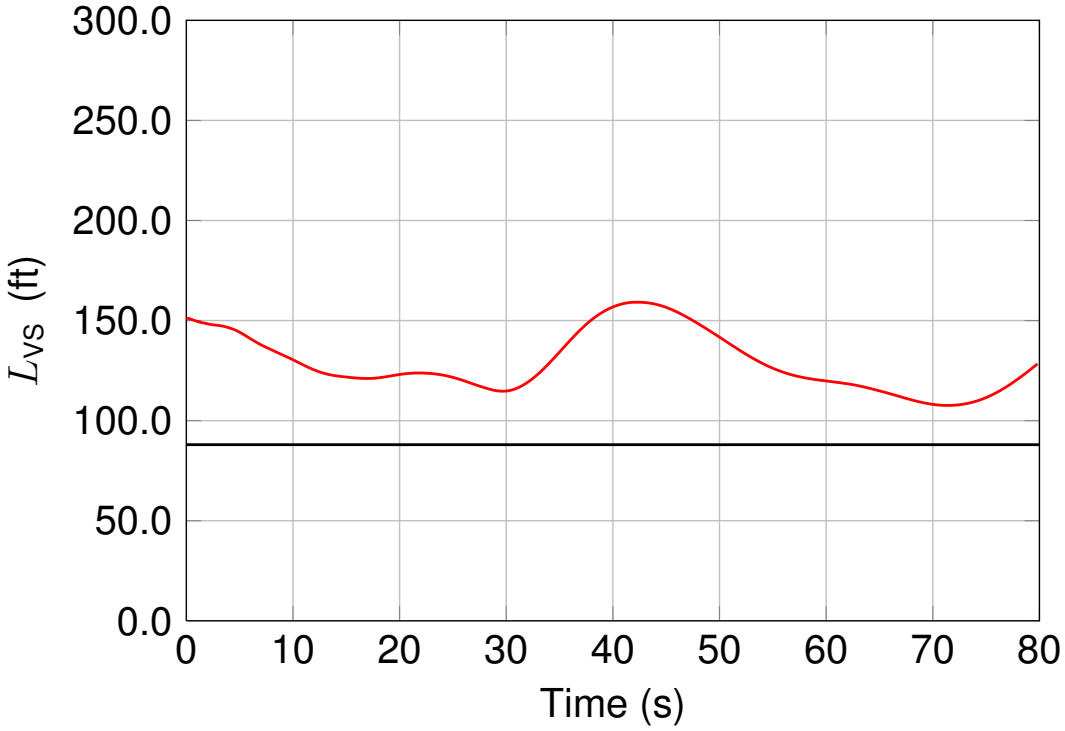


Figure 4.9: Vent-separation distance for 2-MP with PAC.

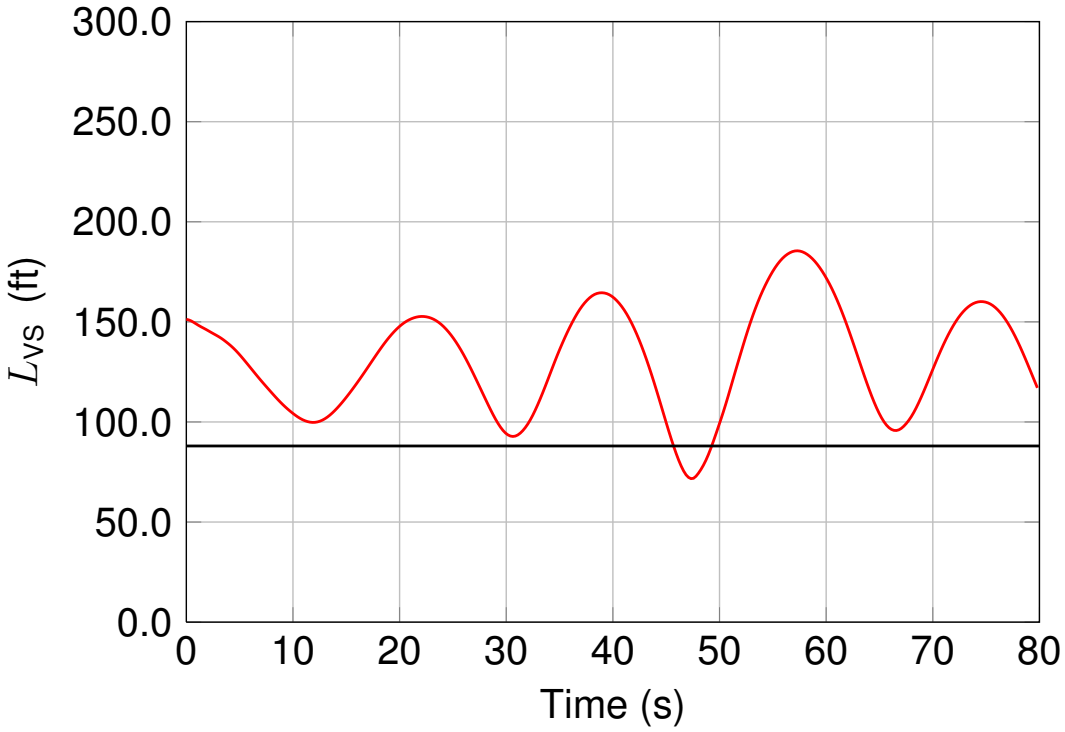


Figure 4.10: Vent-separation distance for 2-MP with PTE.

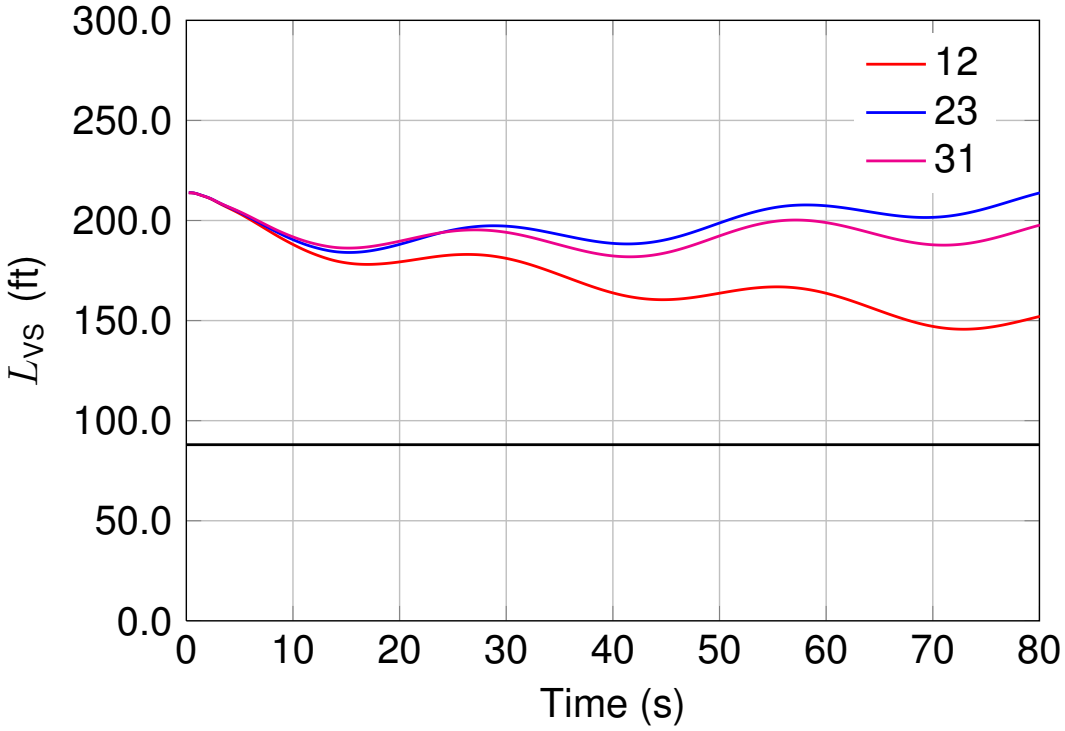


Figure 4.11: Vent-separation distances for 3-MP with PAC.

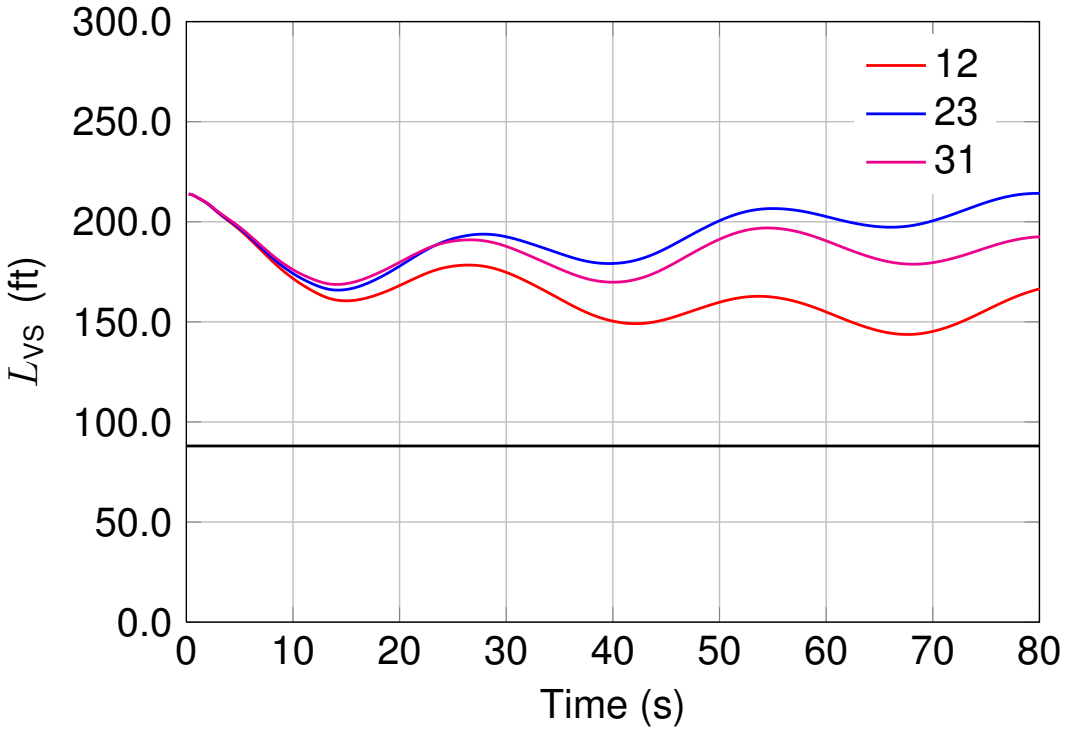


Figure 4.12: Vent-separation distances for 3-MP with PTE.

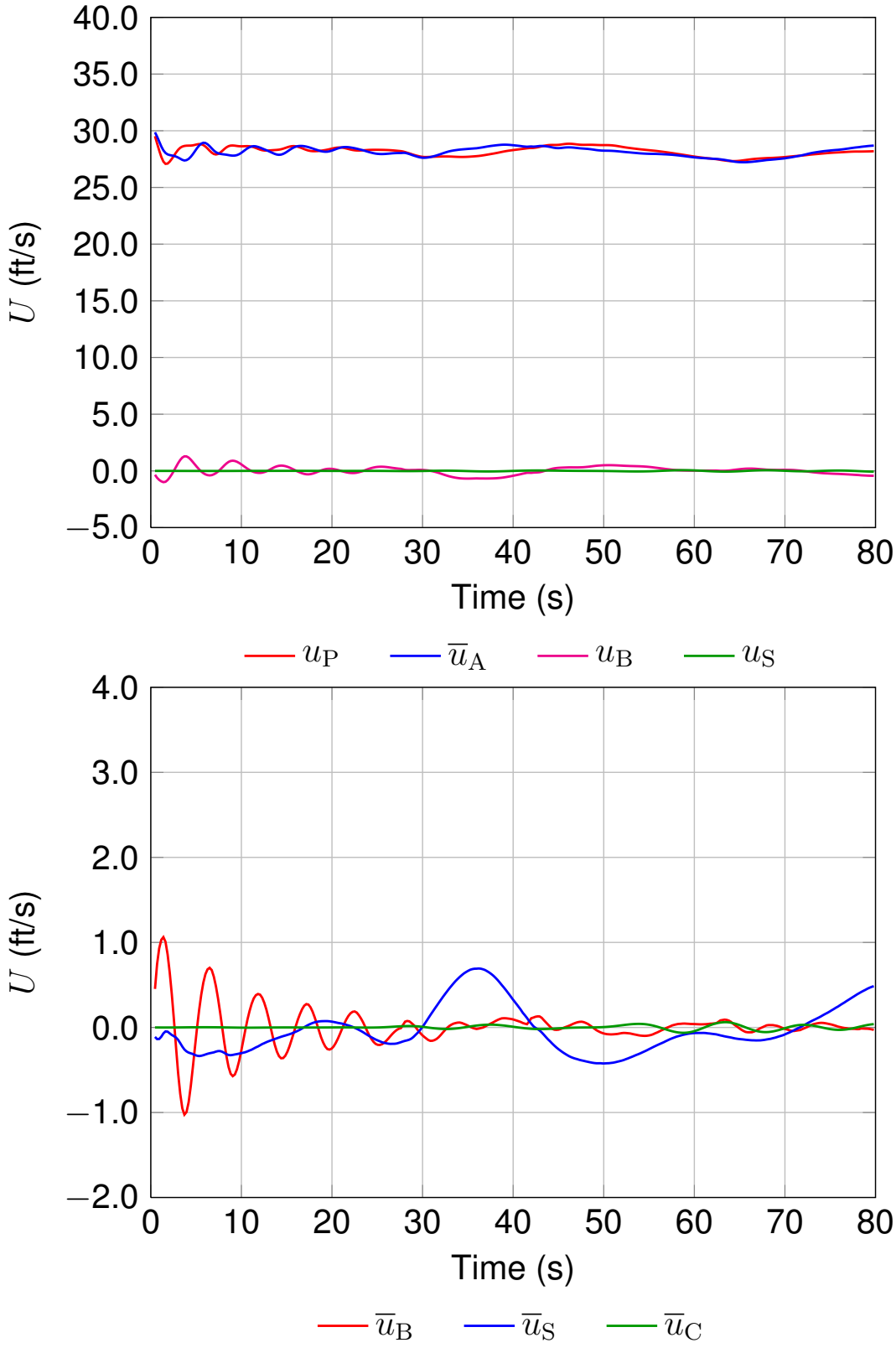


Figure 4.13: Decomposition of the descent speed for 2-MP with PAC.

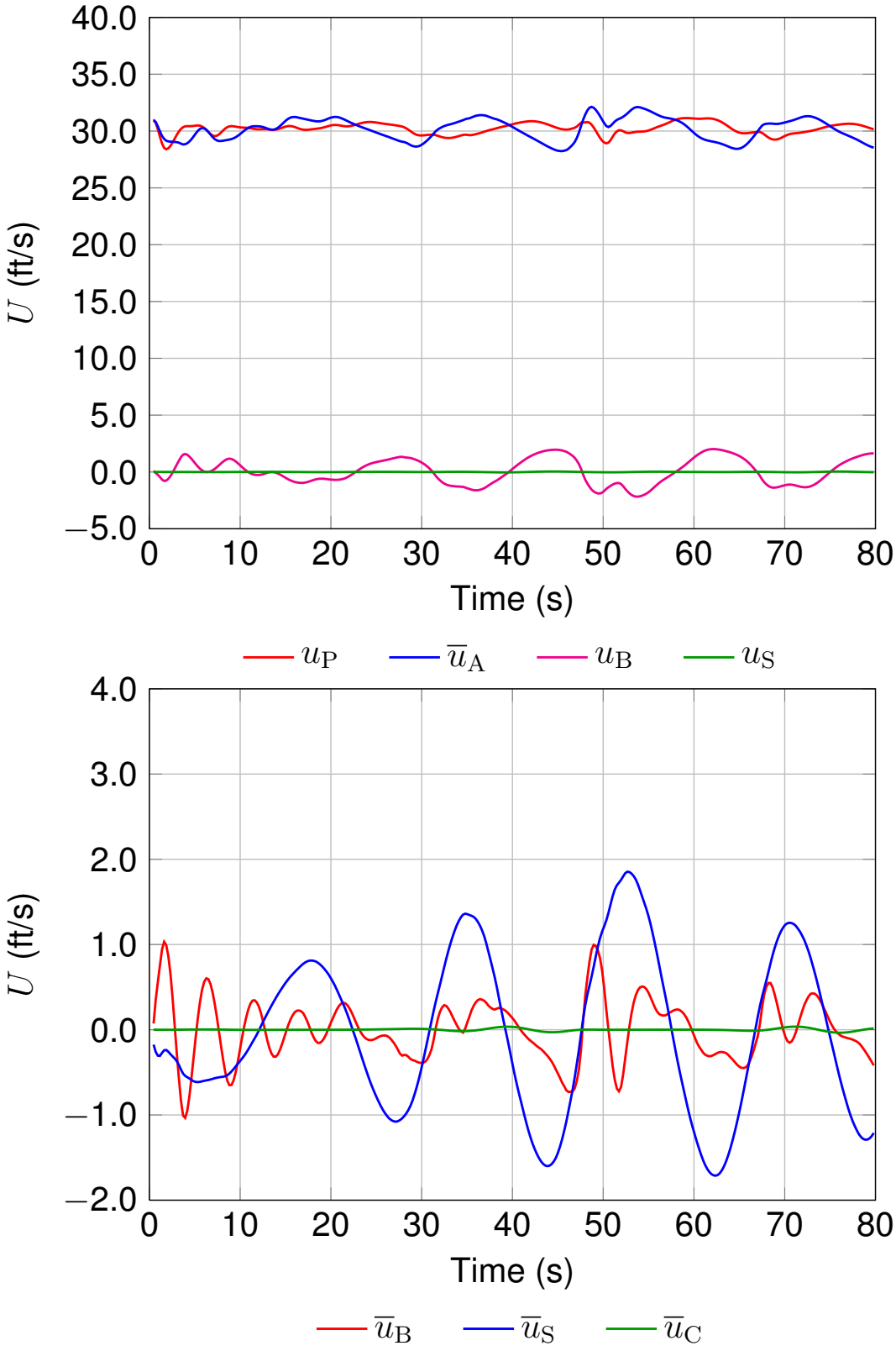


Figure 4.14: Decomposition of the descent speed for 2-MP with PTE.

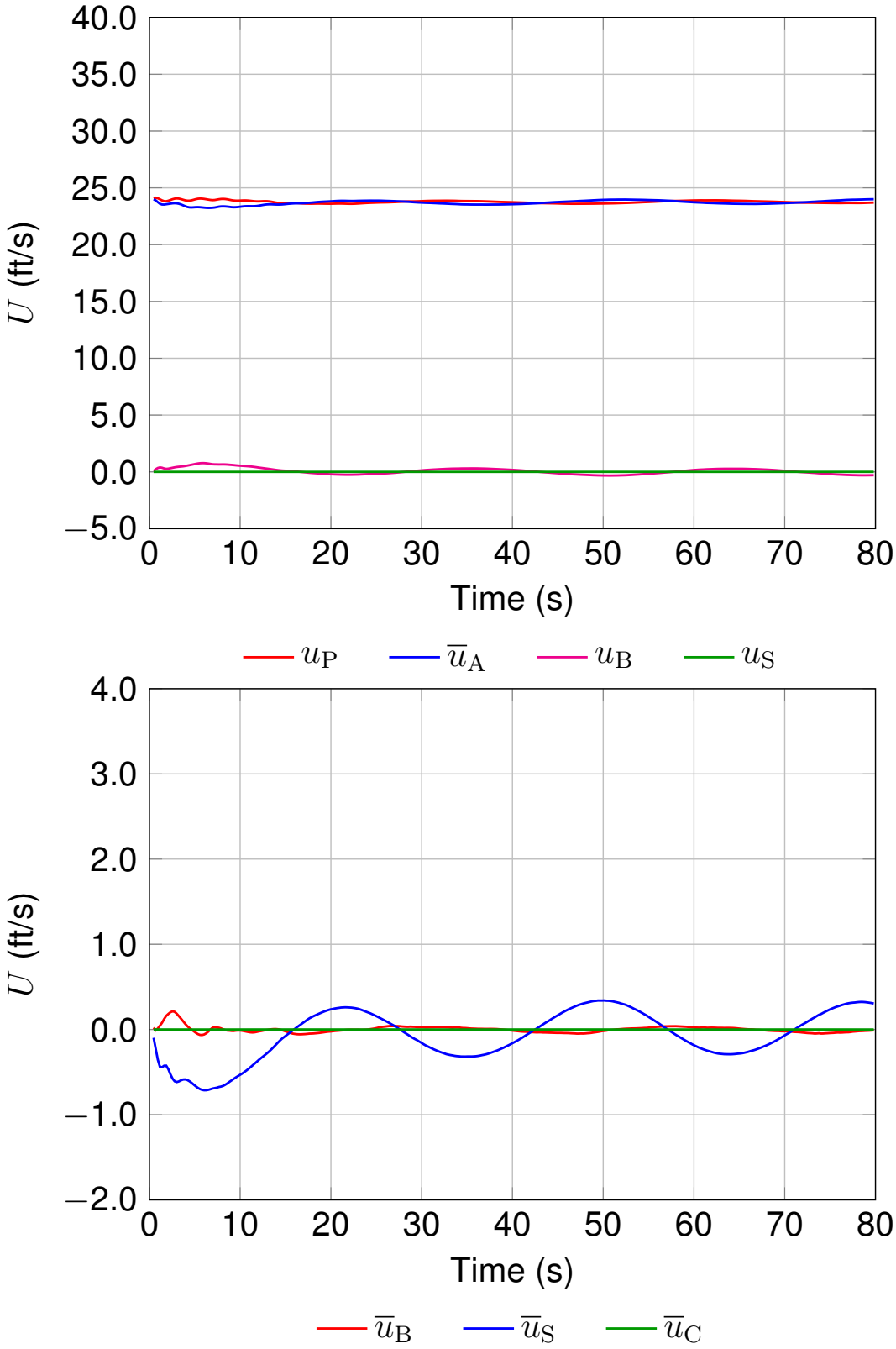


Figure 4.15: Decomposition of the descent speed for 3-MP with PAC.

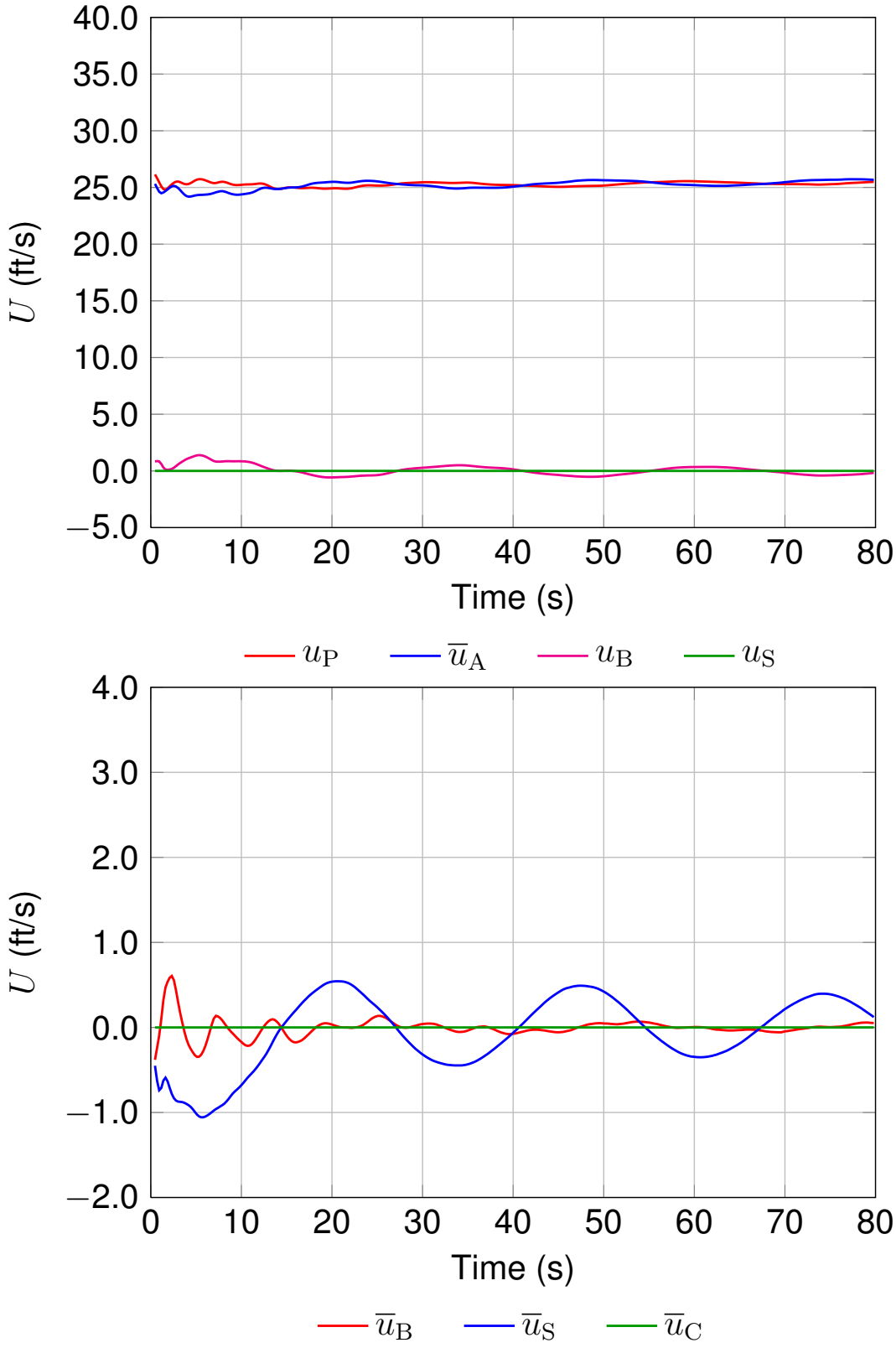


Figure 4.16: Decomposition of the descent speed for 3-MP with PTE.

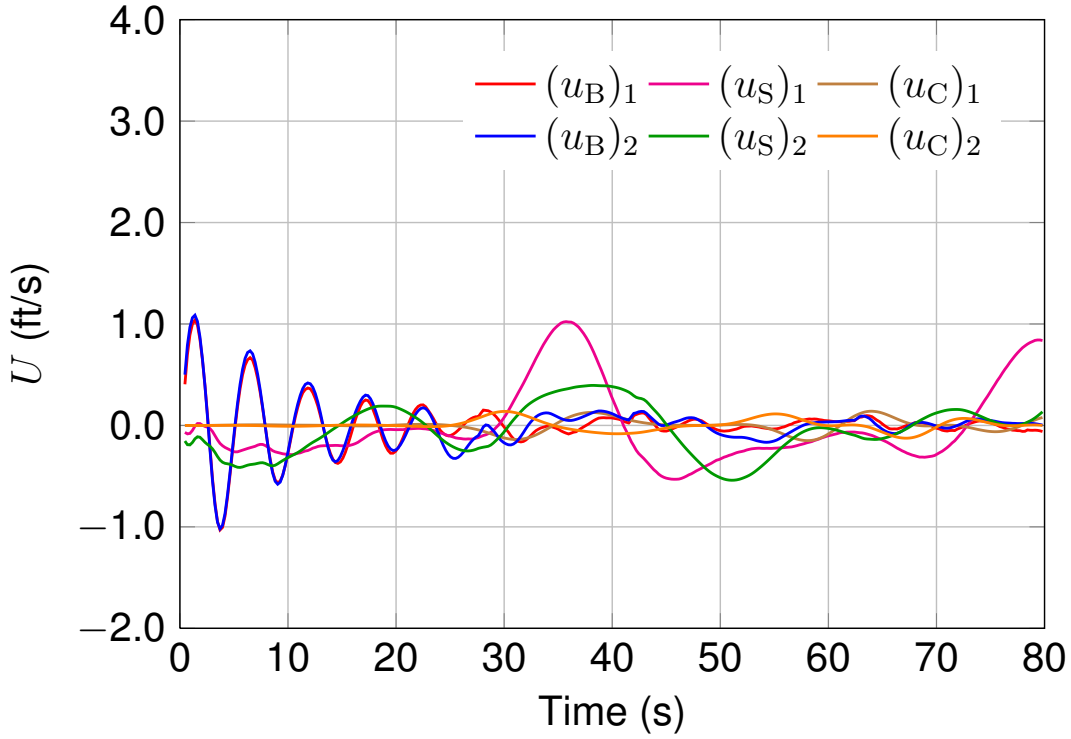


Figure 4.17: Individual-parachute contributions to descent speed for 2-MP with PAC.

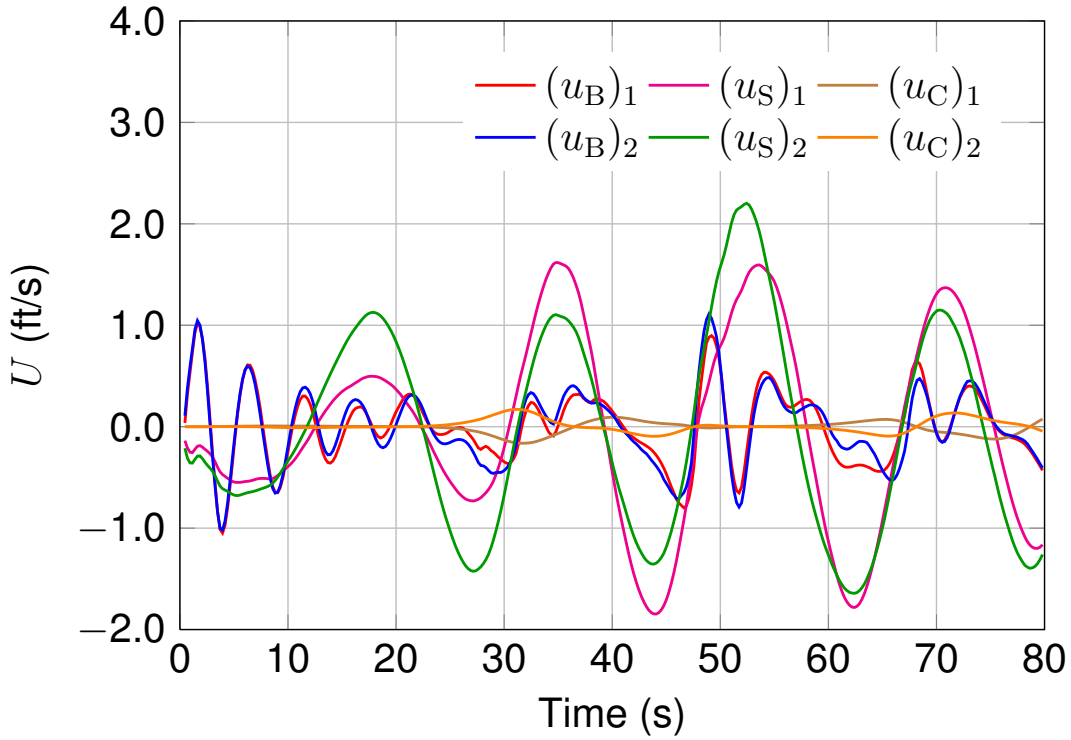


Figure 4.18: Individual-parachute contributions to descent speed for 2-MP with PTE.

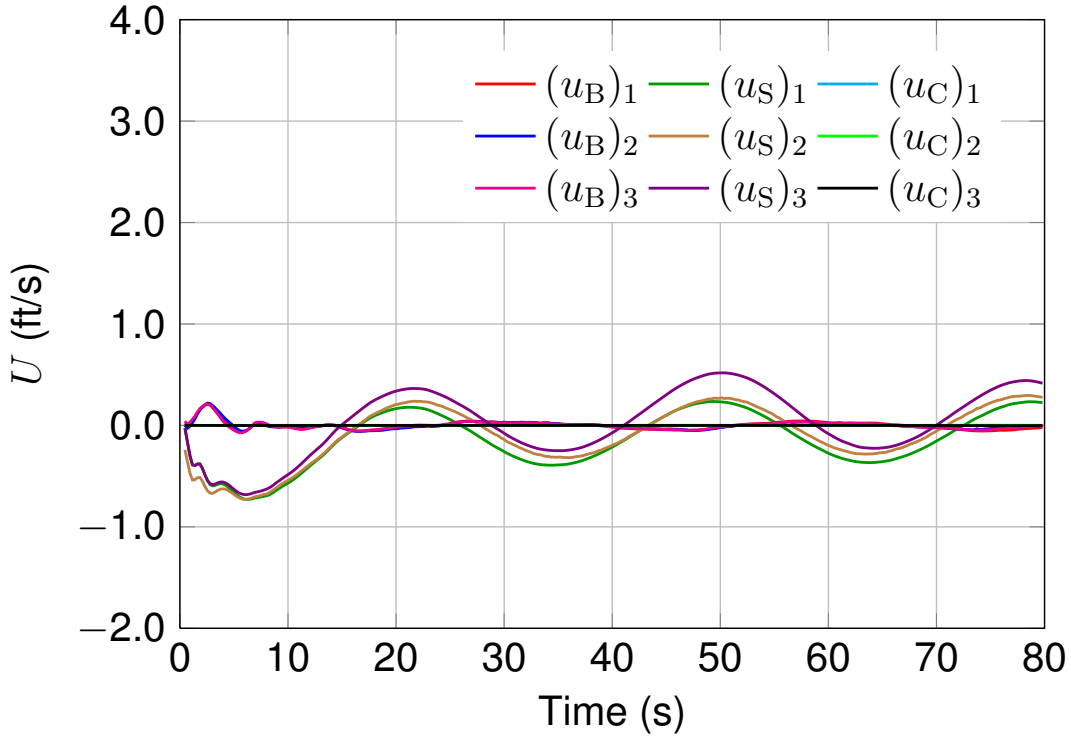


Figure 4.19: Individual-parachute contributions to descent speed for 3-MP with PAC.

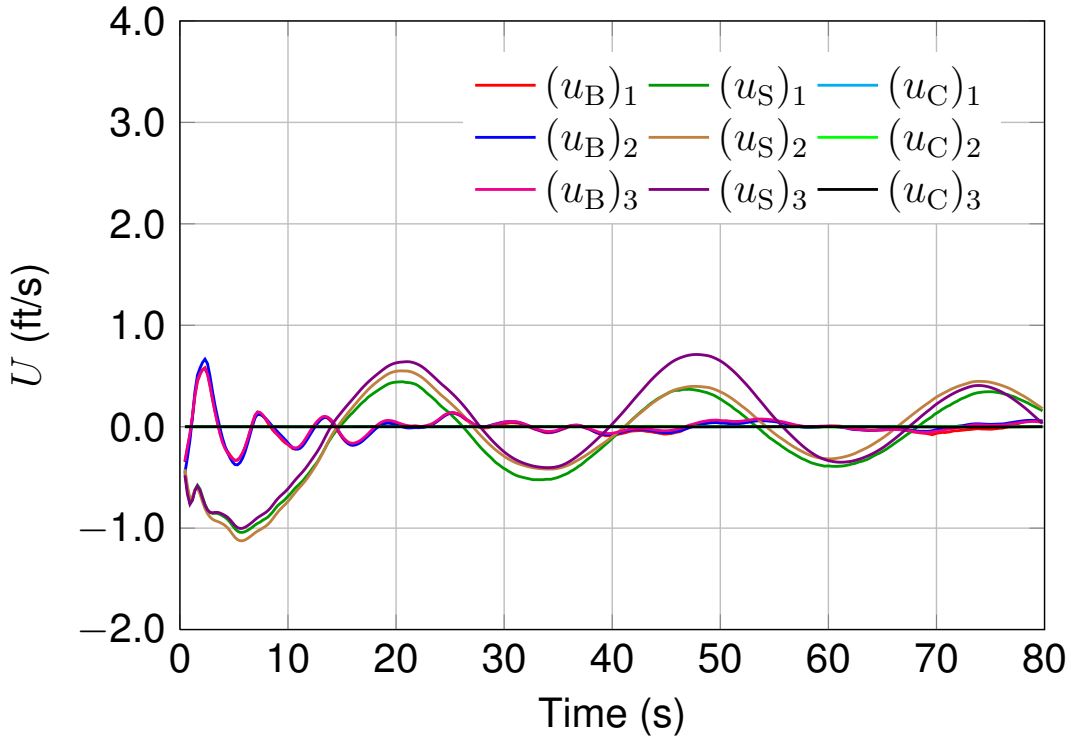


Figure 4.20: Individual-parachute contributions to descent speed for 3-MP with PTE.

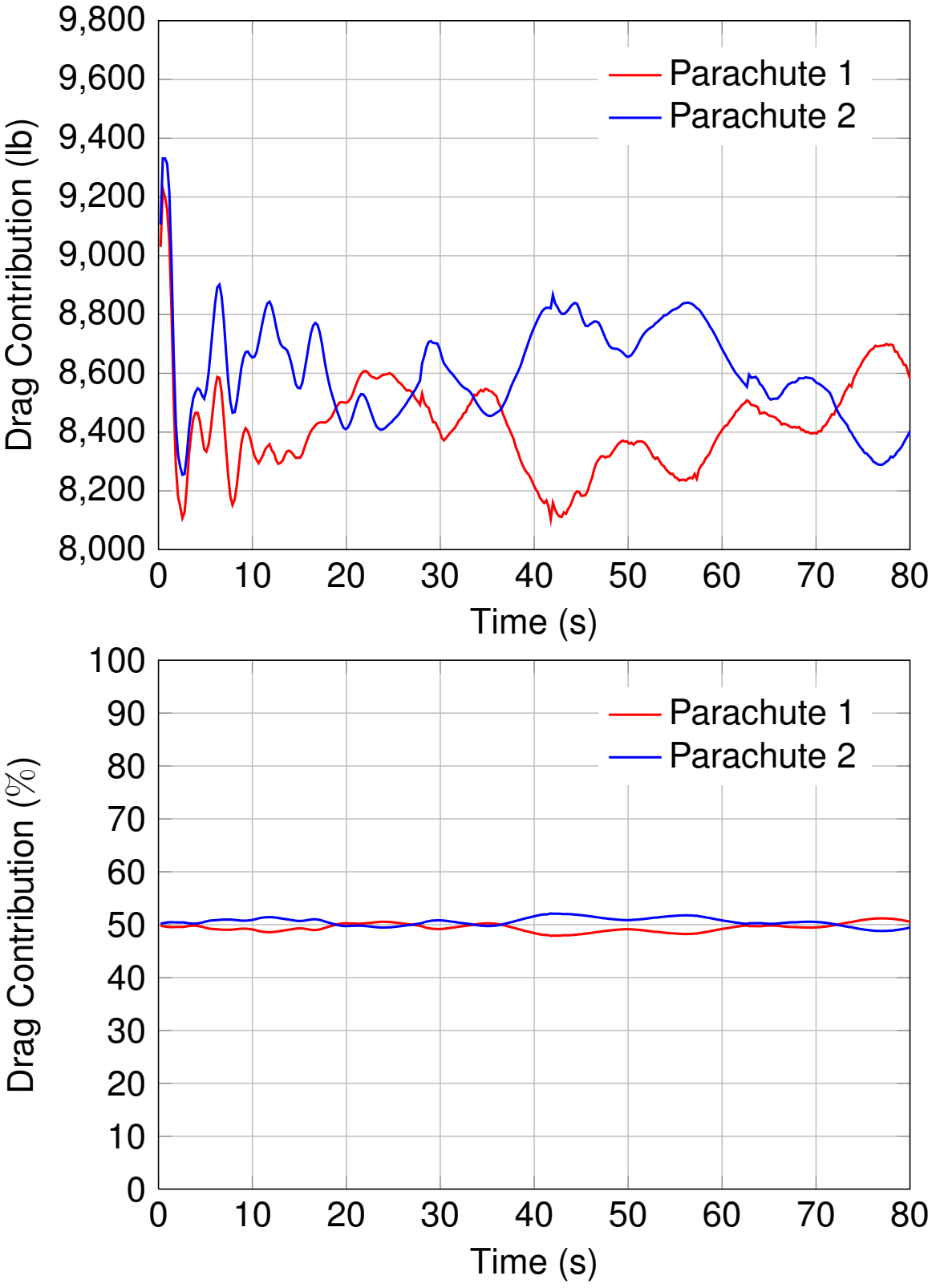


Figure 4.21: Individual-parachute contributions to drag for 2-MP with PAC.

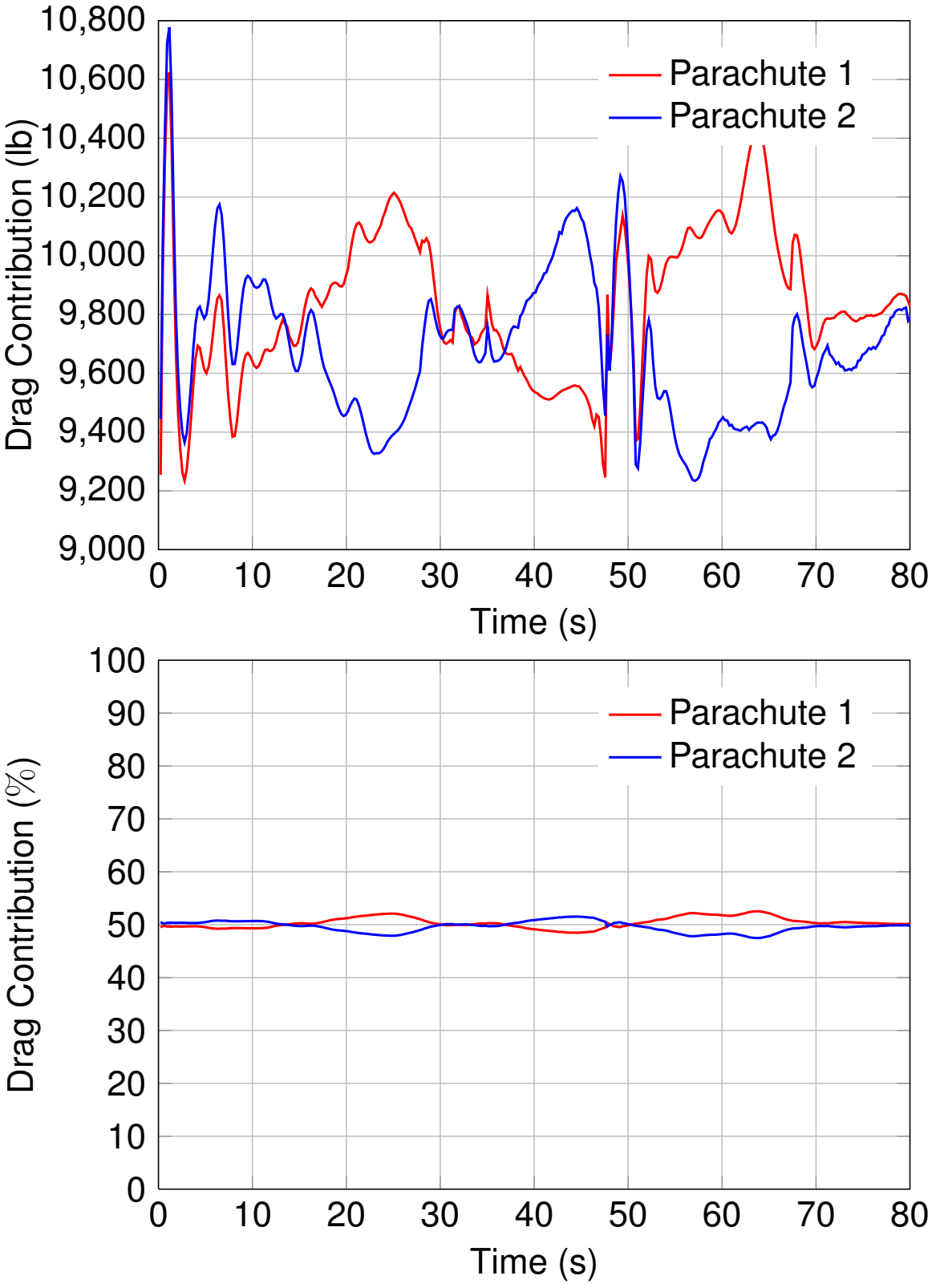


Figure 4.22: Individual-parachute contributions to drag for 2-MP with PTE.

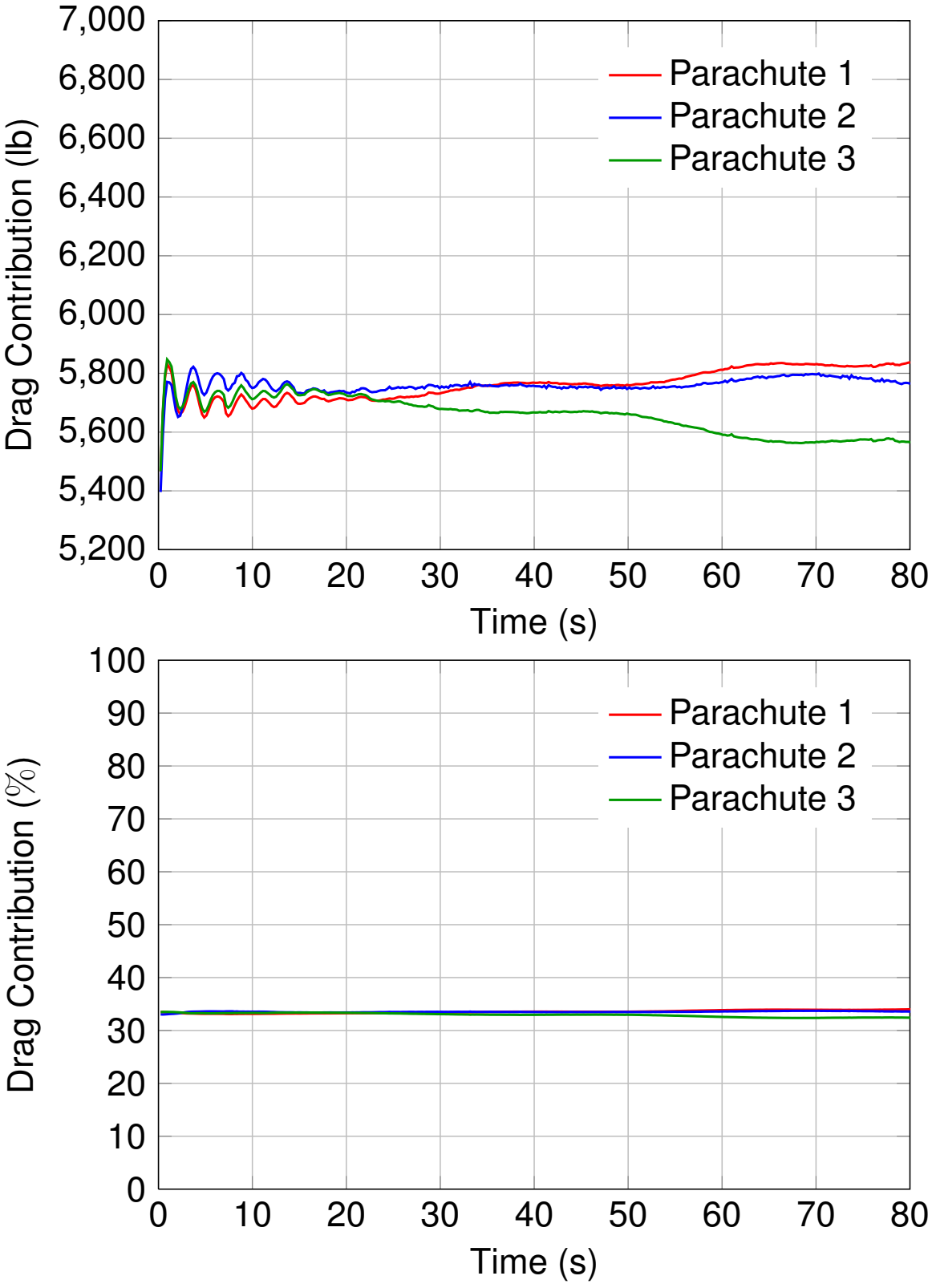


Figure 4.23: Individual-parachute contributions to drag for 3-MP with PAC.

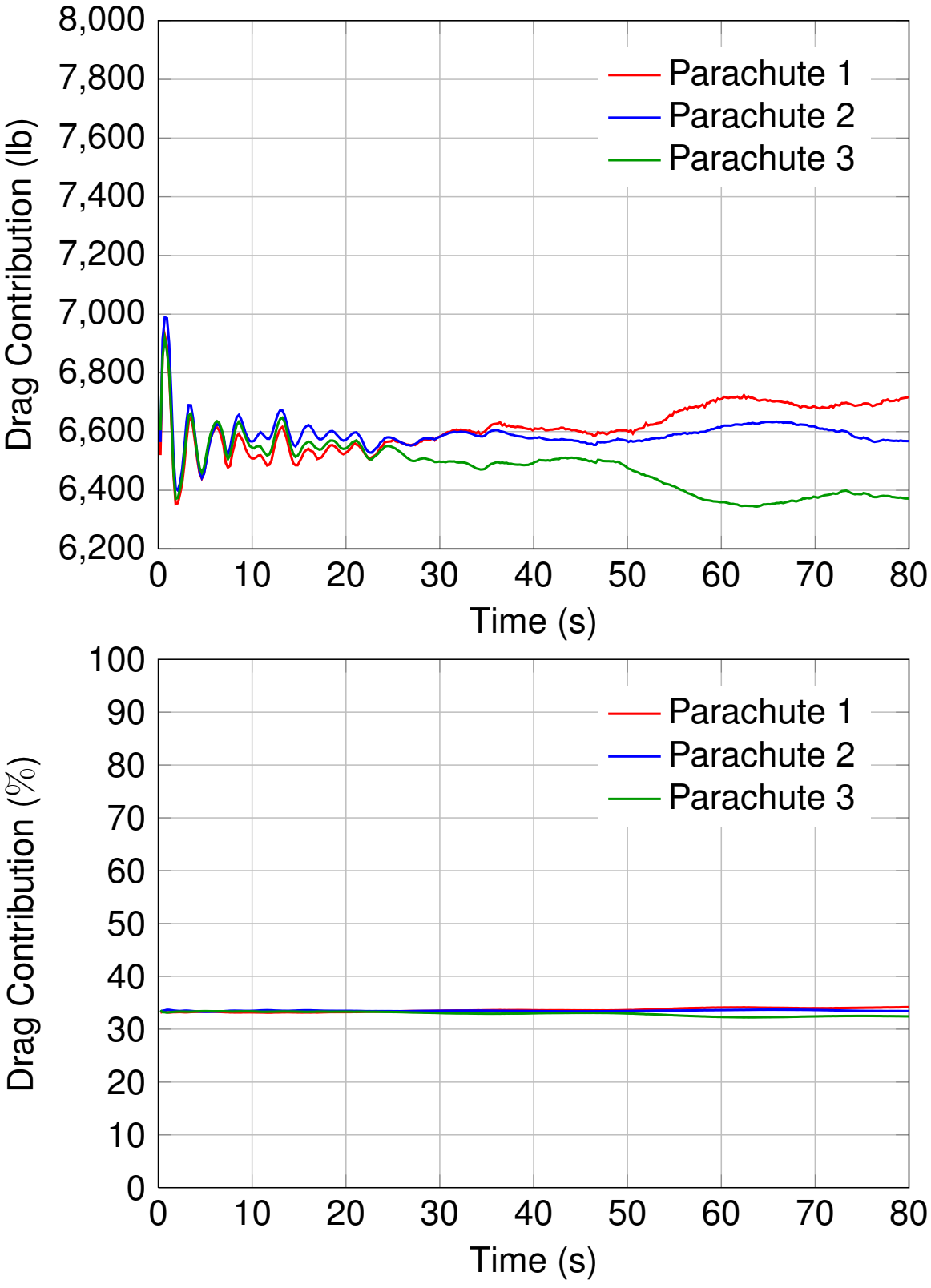


Figure 4.24: Individual-parachute contributions to drag for 3-MP with PTE.

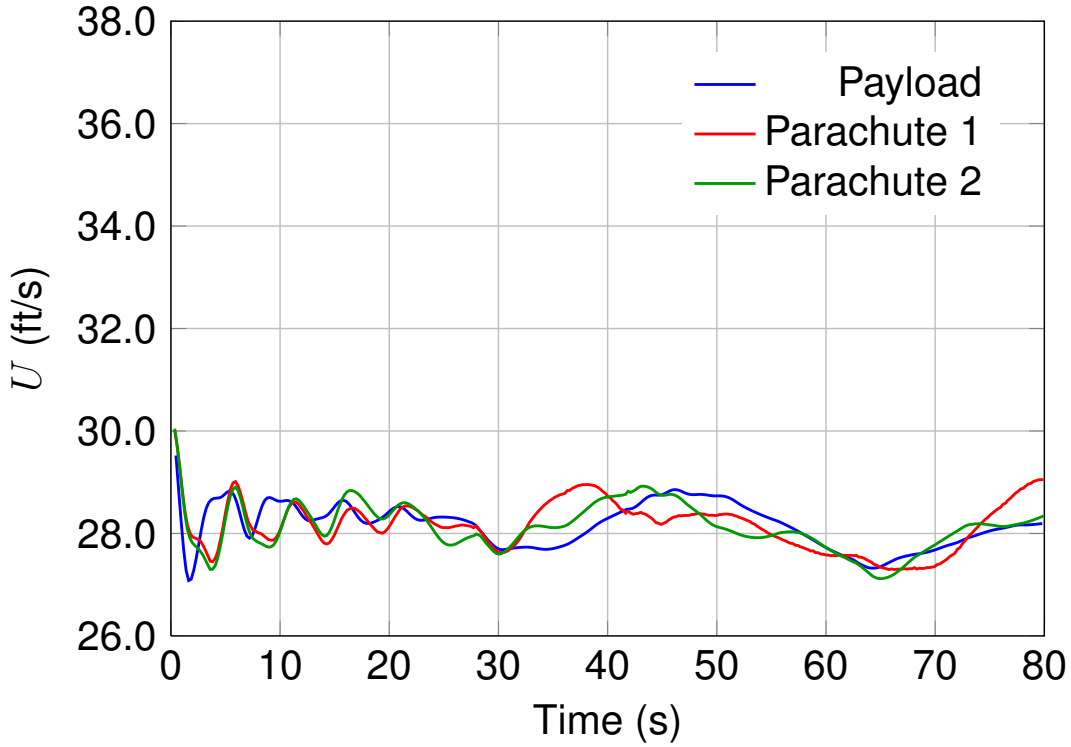


Figure 4.25: Payload and canopy-centroid descent speeds for 2-MP with PAC.

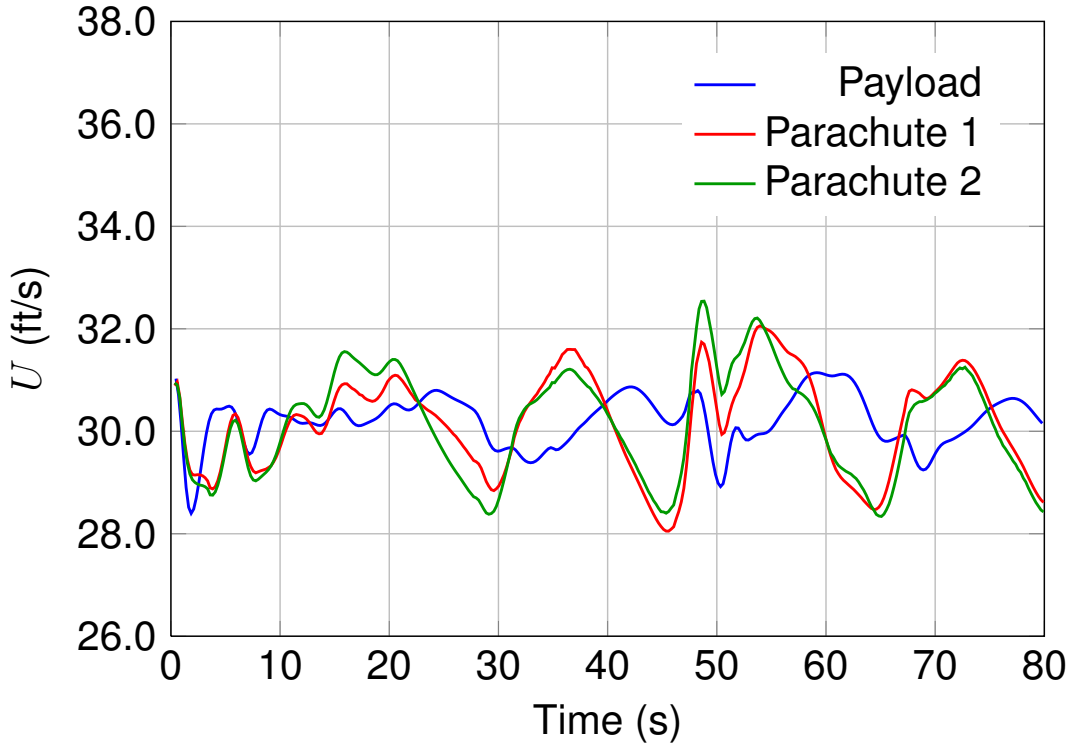


Figure 4.26: Payload and canopy-centroid descent speeds for 2-MP with PTE.

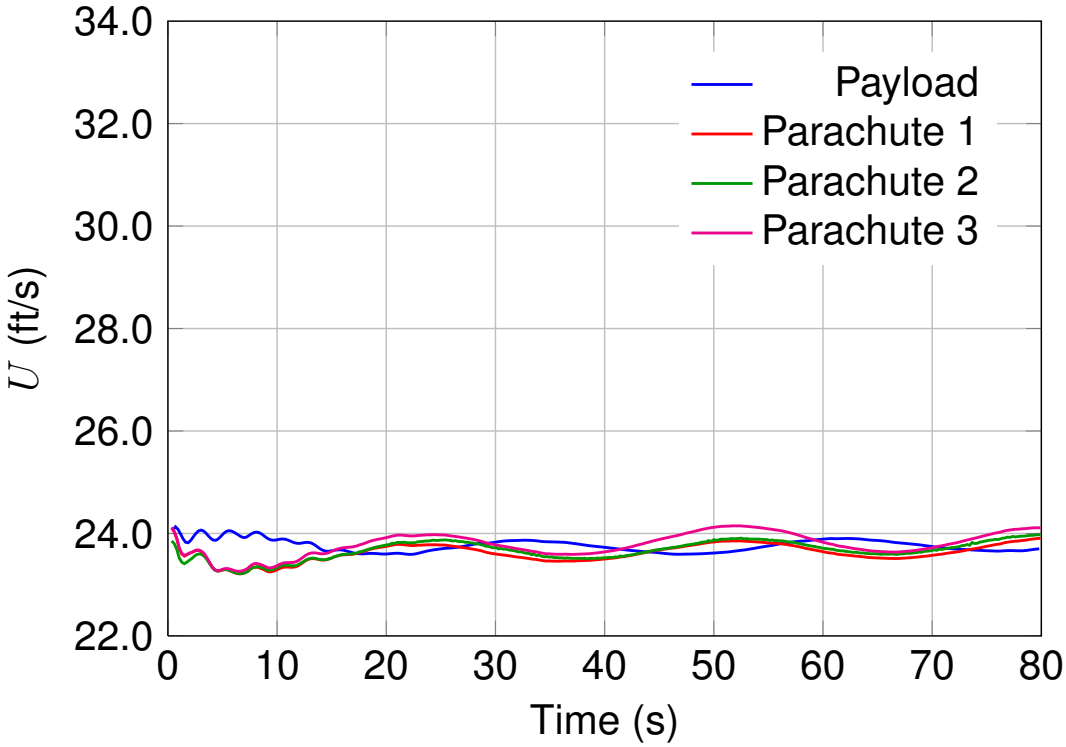


Figure 4.27: Payload and canopy-centroid descent speeds for 3-MP with PAC.

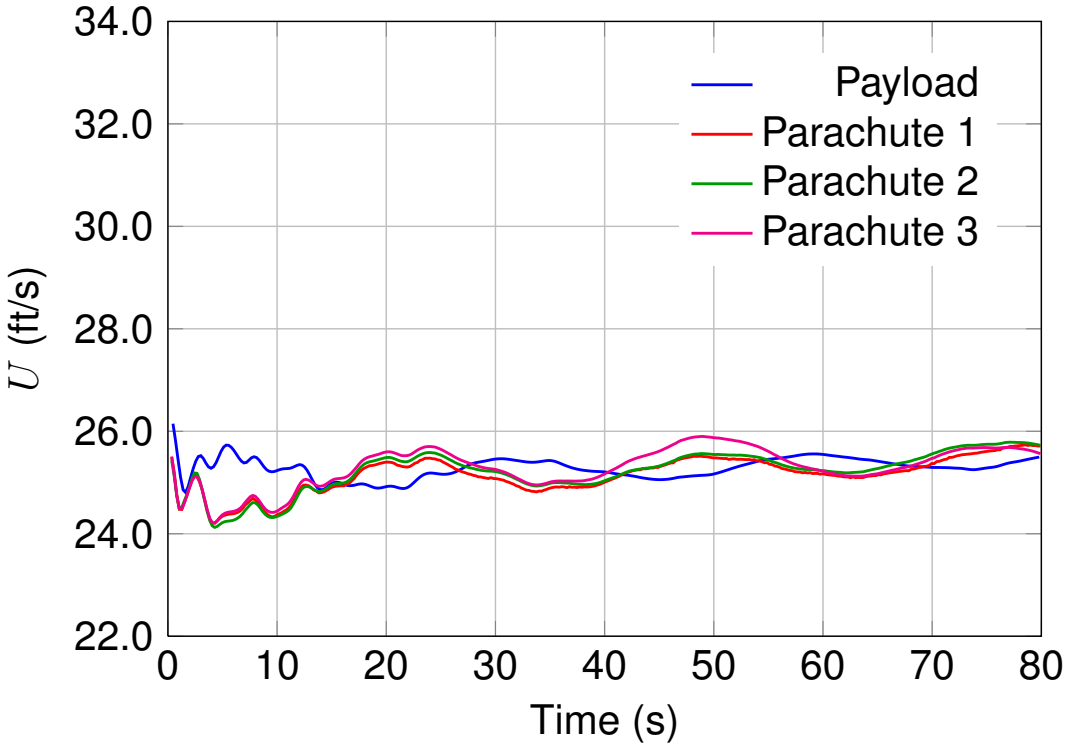


Figure 4.28: Payload and canopy-centroid descent speeds for 3-MP with PTE.

4.2 Test Computations with Single MP Parachutes

The single modified-porosity ringsail parachute presented in this thesis is based off of a design by the Japan Aerospace Exploration Agency (JAXA). Specifically, it is designed to be a scaled down wind-tunnel model currently undergoing empirical tests. First, we present several case studies for the subscale version of the JAXA parachute. We also present an investigation of HMGP porosity.

4.2.1 Single Subscale MP Parachute

The subscale JAXA ringsail parachute has a nominal diameter of 730 mm, and has 40 gores, 3 rings, and 6 sails. Similar to the NASA modified-porosity ringsail parachute described in the previous section, the subscale design features a missing sail and 8 "windows." The windows are created by removing every 5th gore from Sail 8. The ratio L_s/D_o is 1.54. The material properties are the same as those used for the PA parachute computations in [60], with a few exceptions. The canopy membrane properties are uniform, and are the same as MP Sails 4–9. In addition, the suspension lines have diameter 1.5×10^{-3} m and the riser has diameter 2.5×10^{-2} m. A summary of the cases computed is shown in Table 4.3. For the unpinned case, we use a payload weight of 47.5 N.

Table 4.3: Summary of cases for the subscale JAXA parachute. The reference speed is in m/s.

Case	U_{ref}	Payload	Flow
JAXA-1	18	pinned	vertical
JAXA-2	18	unpinned	vertical
JAXA-3	18	pinned	horizontal
JAXA-4	15	pinned	vertical

Starting Conditions

The unstressed single parachute shape is constructed from a quarter sphere such that its nominal diameter is 730 mm. It is then inflated to a stressed condition by applying a uniform pressure differential of 198.5 N/m^2 applied to each node in the parachute canopy. This is the stagnation pressure.

After that, we generate a fluid mechanics mesh and do a fluid mechanics computation, holding the structural parachute shapes and positions fixed. The inflow velocity is 18 m/s. We use this result as the starting condition for the symmetric FSI computation. For the symmetric FSI computation, the payload node is fixed. After the structure settles into a periodic breathing motion in symmetric FSI (about 4,000 time steps), the four different cases are begun.

JAXA-1 continues into full FSI by desymmetrizing over 300 time steps. JAXA-2 continues for 300 time steps in symmetric FSI after the payload condition is changed from pinned to unpinned. After symmetric FSI, it is desymmetrized over 300 time steps and continues into full FSI. JAXA-3 has the gravity vector changed from the vertical direction to the horizontal direction instantaneously as it starts 300 time steps of desymmetrization to full FSI. JAXA-4 is returned to a fluid-only computation with the structural parachute shapes and positions fixed at about halfway through the breathing cycle. The inflow velocity is ramped down from 18 m/s to 15 m/s in 300 time steps using a cosine form. After that we compute an additional 500 time steps followed by full FSI computations, skipping the symmetric FSI and desymmetrization stages.

Computational Methods and Parameters

The computational domain is cylindrical with a diameter of 11 m and a height of 10 m. Figure 4.29 shows, for a single parachute, the canopy structure mesh and the fluid mechanics interface mesh. The fluid mechanics volume mesh consists of four-node

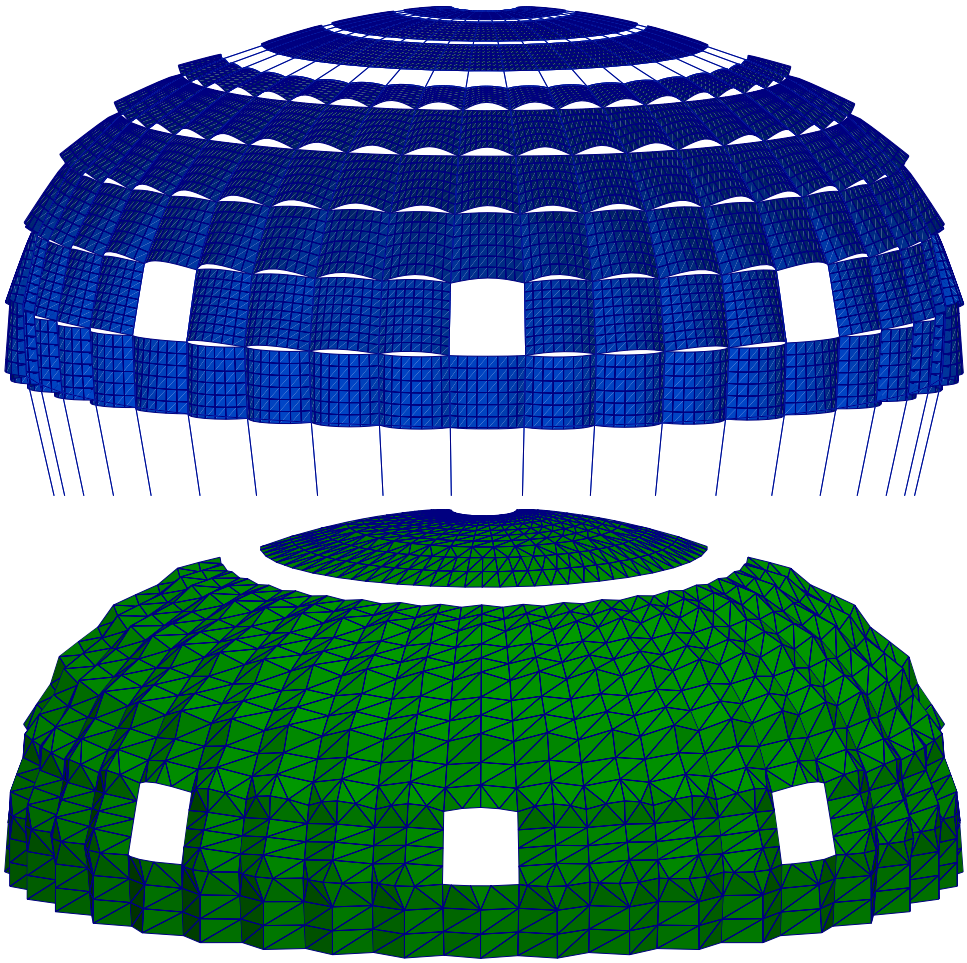


Figure 4.29: Structural mechanics mesh (top) and fluid interface mesh (bottom) for a subscale MP parachute. For the number of nodes and elements at these interfaces, see Table 4.4.

tetrahedral elements, and the membrane elements used in the parachute structure are quadrilateral. The number of nodes and elements for the fluid and structure are given in Table 4.4. We move the reference frame with a vertical velocity of U_{ref}, and translate the mesh horizontally and vertically with the average displacement rate of the structure beyond the reference velocity U_{ref}. We use the velocity form of the free-stream conditions at the lateral boundaries as well since the mesh translates horizontally.

All computations are carried out in a parallel computing environment. The meshes are partitioned to enhance the parallel efficiency of the computations. Mesh parti-

Table 4.4: Number of nodes (nn) and elements (ne) for the subscale parachute.

	nn	ne
Structure		
Membrane	20,778	22,770
Cable	4,762	5,121
Payload	1	1
Fluid		
Volume	117,956	681,344
Interface	2,356	4,340

tioning is based on the METIS [29] algorithm. In solving the linear equation systems encountered at every nonlinear iteration, the GMRES search technique [36] is used with a diagonal preconditioner.

The structure-only computations are carried out over 31,600 time steps with a time–step size of 0.01 s and 4 nonlinear iterations per time step. The number of GMRES iterations per nonlinear iteration is 100. Mass-proportional damping is 9,863 s^{-1}.

The fluid mechanics computations with fixed shapes and positions are done in two parts. The first part uses the semi-discrete formulation given in [65]. We compute 2,000 time steps with a time-step size of 2.03×10^{-3} s and 6 nonlinear iterations per time step. The number of GMRES iterations per nonlinear iteration is 90. There is no porosity. The second part uses the DSD/SST-TIP1 technique, with the same SUPG test function option and stabilization parameters as those described in Section 4.1.2. We compute 4,500 time steps with a time-step size of 2.03×10^{-4} s, 6 nonlinear iterations per time step, and 90 GMRES iterations per nonlinear iteration. The porosity model is HMGP-FG [52]. The HMGP coefficients k_{F} and k_{G} are derived from the coefficients used for the NASA PA design (see Table 4.5). We use the same k_{F} value for all patches, since the entire membrane is assumed to have the same material properties as Sails 4-9 for the PA parachute. Specifically, we use the $\mathrm{k_F}$ value from Patch 5 of the PA parachute. The k_{G} value for the sail slits matches Patch 9 since the same leading edge fullness is assumed for all JAXA sails. The $\mathrm{k_G}$ value for the ring gaps

basically matches Patch 3. Table 4.6 shows the k_F and k_G values.

Table 4.5: HMGP coefficients k_F and k_G used in the PA parachute computations reported in [60, 54, 53] (note that Patch 1 and Patch 14 do not have a k_G value because those patches do not contain a ring gap or sail slit). Starting from these values, we generate the k_F and k_G values for the JAXA parachute (see Table 4.6).

Patch	k_F (CFM)	k_G
1	125.5	
2	115.0	0.967
3	115.0	0.971
4	75.8	0.960
5	39.9	0.949
6	39.8	0.756
7	39.7	0.769
8	39.7	0.824
9	39.6	0.831
10	68.8	0.816
11	98.3	0.819
12	97.7	0.820
13	97.0	0.867
14	97.1	

In the symmetric FSI computation we use the SSTFSI-TIP1 technique, with the SUPG test function option WTSA. The stabilization parameters used are those described in Section 4.1.2. The porosity model is HMGP-FGR. The interface projection methods used include the SSP technique. The fully-discretized, coupled fluid and structural mechanics and mesh-moving equations are solved with the quasi-direct coupling technique. We use selective scaling (see [75]), with the scale for the structure part set to 10 and for the other parts set to 1. The time-step size is 2.03×10^{-4} s, with 6 nonlinear iterations per time step. The number of GMRES iterations per nonlinear iteration is 90 for the fluid+structure block, and 30 for the mesh-moving block.

In the FSI computations we use the SSTFSI-TIP1 technique, with the same SUPG test function option and stabilization parameters as those described above. The porosity model is HMGP-FGR. The interface projection methods used include the SSP technique. We use the quasi-direct coupling technique and selective scaling, with

Table 4.6: HMGP coefficients k_F and k_G used for the JAXA parachute computations. The k_F values are derived from a PA sail patch with the same fabric porosity, and k_G is estimated from a similar PA gap and slit with with same leading edge fullness (note that Patch 4 and Patch 5 do not have a k_G value because those patches do not contain a ring gap or sail slit).

Patch	k_F (CFM)	k_G
1	39.9	
2	39.9	0.970
3	39.9	0.970
4	39.9	
5	39.9	
6	39.9	0.831
7	39.9	0.831
8	39.9	0.831
9	39.9	0.831
10	39.9	0.831
11	39.9	

the scale for the structure part set to 1,000 and for the other parts set to 1. The time-step size is 2.03×10^{-4} s, and the number of nonlinear iterations per time step is 6. The number of GMRES iterations per nonlinear iteration is 180 for the fluid+structure block, and 30 for the mesh-moving block. We note that for the JAXA-2 case, we had to increase the number of GMRES iterations for the fluid+structure block to 360 at the start of the FSI computations.

Preliminary Results

Figure 4.30 shows the fluctuations in the distance between the canopy centroid and payload. Only the JAXA-2 case shows any noticeable change in this distance over time. The change in the JAXA-2 length correponds with the changes in parachute diameter, seen in Figure 4.31, which is due to breathing. This correlation, along with the negligible change in canopy centroid to payload distance for all of the other cases, suggests that the stretch in the riser does not impact the payload descent speed as much as the breathing motion of the parachute. Figure 4.32 shows the

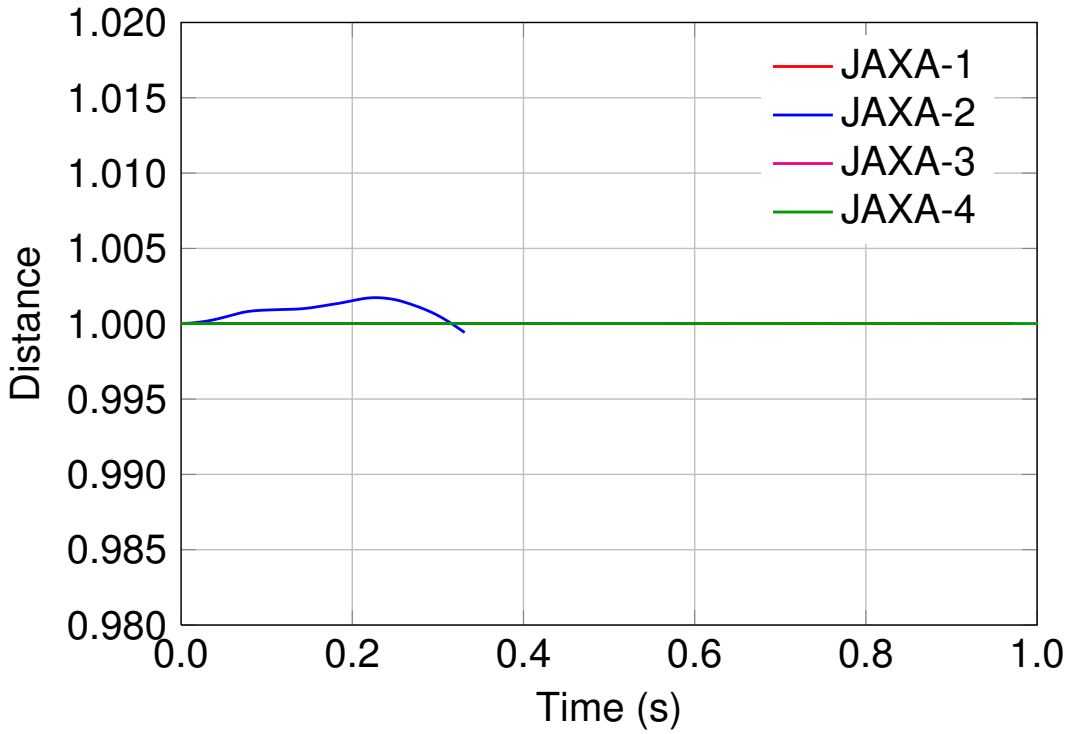


Figure 4.30: Distance from the canopy centroid to the payload normalized by the JAXA-1 average distance.

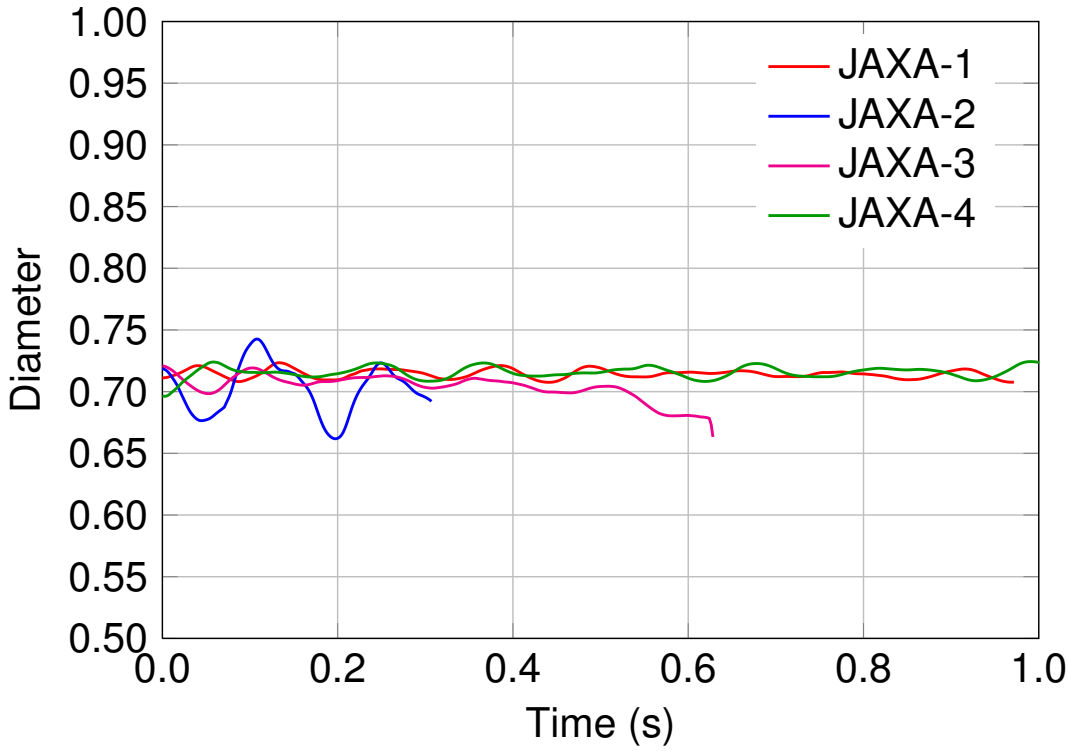


Figure 4.31: Average diameter of the canopy normalized by the nominal diameter of the canopy.

drag coefficient for the four subscale JAXA cases. Most obvious in this plot are the much greater fluctuations for the JAXA-2 case. This is further supported by the plot of average canopy diameter in Figure 4.31. This suggests that the pinned condition may mask larger fluctuations in descent speed that would be seen in a drop test of the same parachute design. Figure 4.33 shows the angle between the vertical and

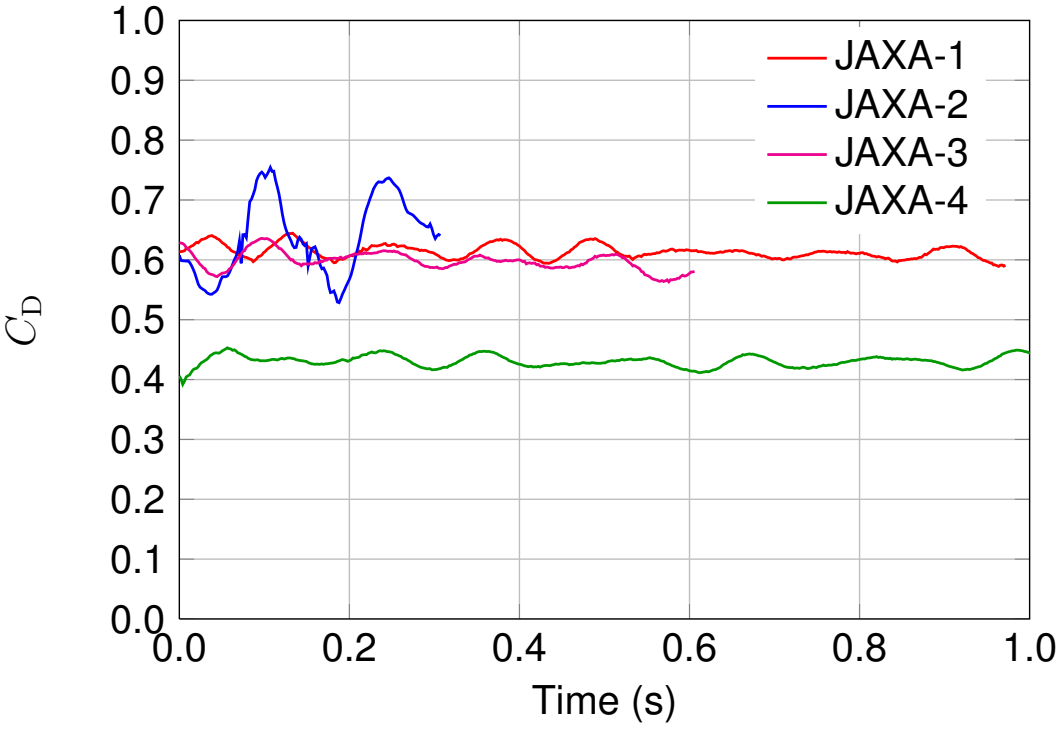


Figure 4.32: Drag coefficient for the subscale JAXA cases.

parachute axes for the JAXA-3 case. This angle is continuously increasing throughout the computation and reaches over 20° by the time the computation is stopped. In this case, we stopped the computation where it required remeshing. As a sample flow visualization, Figure 4.34 shows the velocity vectors for the JAXA-1 case.

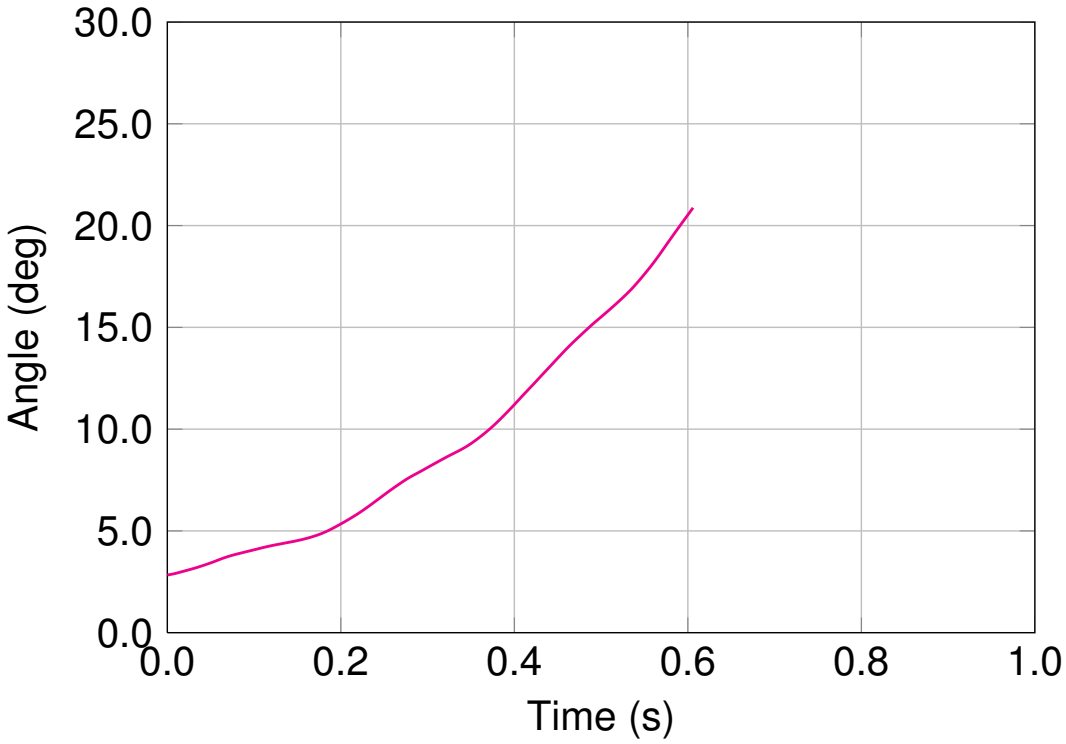


Figure 4.33: Angle between the vertical and parachute axes for the JAXA-3 case.

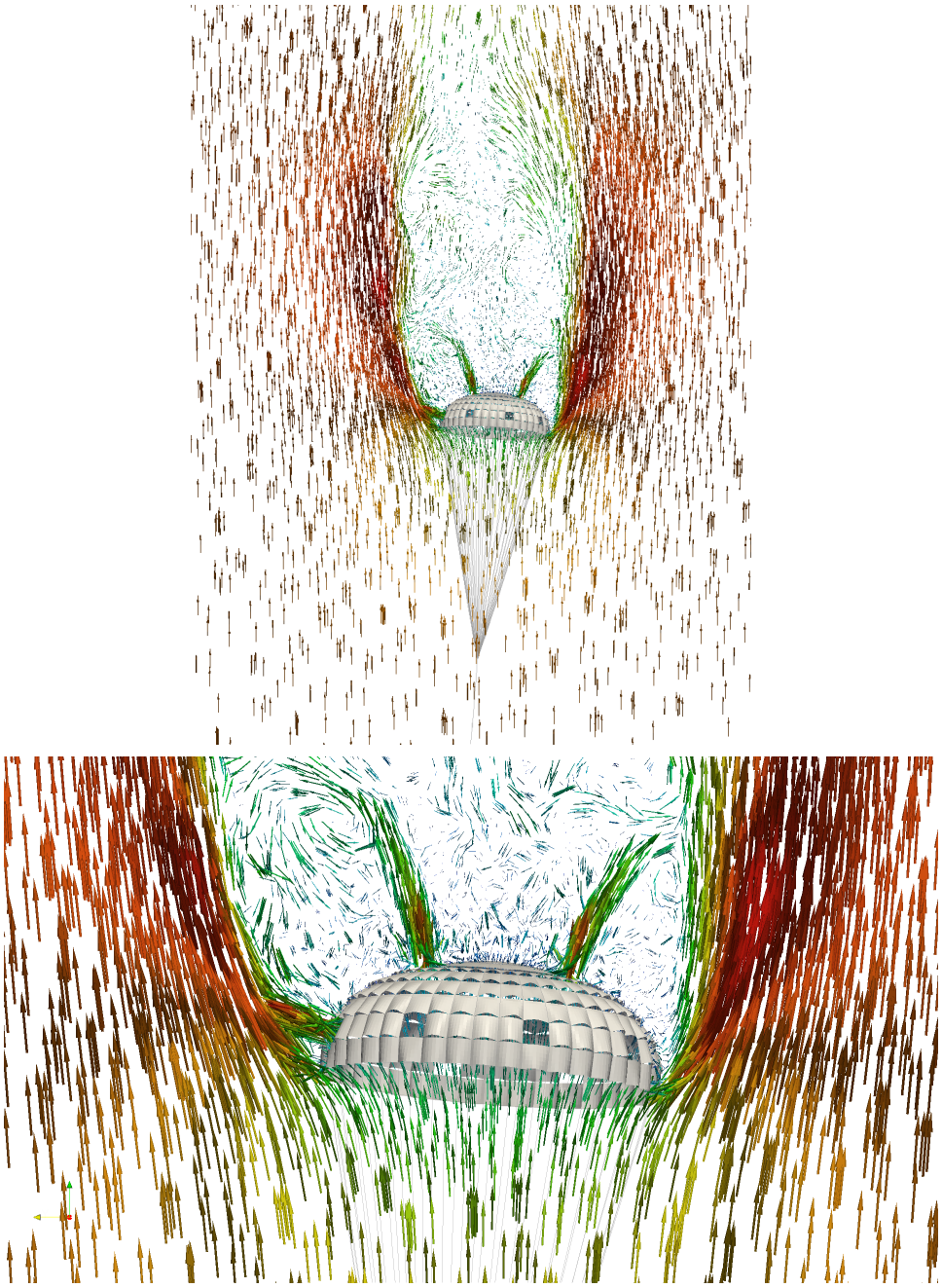


Figure 4.34: Flow field for the JAXA-1 case at t = 0.95 s. Velocity vectors are colored by magnitude.

4.2.2 Periodic *n*-gore Model

Using the techniques in [52, 60], we build a 5-gore slice of the parachute canopy and compute the flow through the membrane, gaps, and slits. We do this for both the subscale model described in the previous section and a full-scale version of the JAXA parachute, with $D_o = 116$ ft. For simplicity, the windows created by removing every 5th gore from Sail 8 are not included in this model. We use the flow rates to calculate k_F and k_G. We then compare the results from the subscale and full-scale computations.

Starting Conditions

A stand-alone structural mechanics computation is carried out with the undeformed, refined structure mesh using the same method as described in Section 4.2.1. Mass-proportional damping is 49,300 s^{-1}. Once the canopy shape reaches a settled solution, the four-node quadrilateral elements of the structure mesh are subdivided to create a fluid interface mesh composed of the three-node triangular elements, thus modeling the actual parachute geometry. All n-gore computations use the same fluid interface mesh refinement, shown in Figure 4.35, which has 237,400 nodes and 459,360 three-node triangular elements for the full, 40-gore canopy.

Next, one gore is extracted from the refined fluid interface and the gaps and slits of this single-gore mesh are meshed with a specified number of triangular elements (red area in Figure 4.35). A surface mesh is then created for the primary lateral boundary, and it is duplicated and rotated one gore slice (9°) to create the secondary lateral boundary. The remaining domain boundaries are then created and the fluid volume mesh is generated. In the full-scale model, the parachute vent and outflow boundary are located 290 ft and 1,450 ft downstream of the inflow boundary, respectively. In the subscale model, those distances are 1.83 m and 9.13 m. The radial boundary is located 290 ft and 1.83 m from the vent center for the full scale and subscale models,

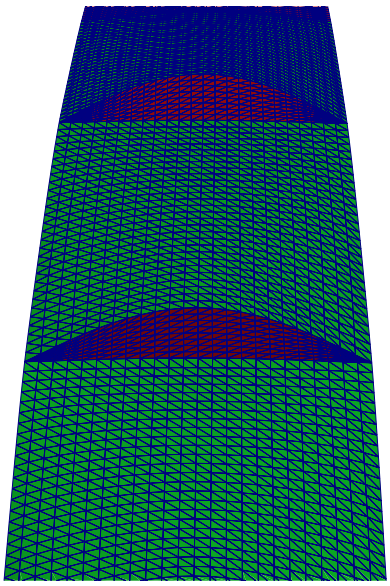


Figure 4.35: One-gore slice of the refined fluid interface mesh.

respectively.

Copies of the single-gore mesh are then rotated at 9° increments and the overlapping boundary nodes are merged to create a n-gore fluid volume mesh, with $n = 5$ in our case. Finally, nodes on the fluid–structure interface are split so that the pressure differential across the canopy can be modeled. The 5-gore model created using this mesh generation method is shown in Figure 4.36. Table 4.7 shows the number of elements and nodes.

Computational Methods and Parameters

The computations are carried out using the DSD/SST-TIP1 technique, with the SUPG test function option WTSA. The stabilization parameters used are those given in Section 4.1.2. The time-step size is 8.75×10^{-5} s for the subscale model and 0.01 s for the full-scale model. The number of nonlinear iterations per time step is 6, and the number of GMRES iterations per nonlinear iteration is 90. Selective scaling is used, with the scale for the pressure part set to 10. The subscale model has a Re value of approximately 9.0×10^{5}, while the full-scale model has Re of 1.8×10^{7}. Properties

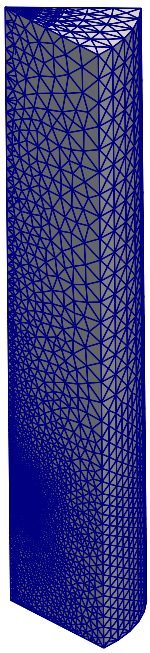


Figure 4.36: Mesh at the external boundaries of the 5-gore model.

of air at standard sea-level conditions are used.

A periodic 5-gore domain, shown in Figure 4.36, is used in these computations. A 5-gore model was selected so that future computations can easily incorporate a window into the n-gore model. All velocity components are specified at the inflow and all stress components are set to zero at the outflow. Rotationally-periodic boundary conditions are used for the lateral boundaries that intersect the canopy. At the radial boundary, we use slip condition in the vertical direction, and no-slip in conditions in the two horizontal directions. A free-stream velocity of 18 m/s is specified at the inflow boundary for the subscale model, and 25 ft/s for the full-scale model. The canopy has no-slip conditions.

Table 4.7: Number of nodes (nn) and elements (ne) for the 5-gore parachute computations with rotationally-periodic boundary conditions.

	nn	ne
Volume	594,226	3,325,095
Interface	29,940	57,420

A stand-alone fluid mechanics computation for each of the two cases is performed using the semi-discrete formulation [65] until the flow field becomes developed and exhibits vortex shedding (about 2,500 time steps). The semi-discrete computations are performed to expedite the development of an initial flow field. The computations are then continued for approximately 3,600 time steps using the DSD/SST-TIP1 formulation. The HMGP-FG porosity coefficients are calculated for each patch using data averaged over the final two cycles of drag oscillations (about 1,200 time steps).

Results

The flow field for each mesh is shown in Figure 4.37, and the calculated fabric and geometric porosity coefficients are given in Table 4.8. Calculated values for k_F and k_G are shown in Figure 4.38. The comparison between the subscale and full-scale JAXA n-gore parachute essentially amounts to a parametric study of Reynolds number effects on the HMGP-FG porosity model. Such a study was performed in prior work [85], and the conclusion was that only differences in Reynolds number of several orders of magnitude have a noticeable effect on k_F and k_G. These computations, only differing in Reynolds number by about an order of magnitude, predictably yield very similar values.

What is most valuable from this computation, however, is that we now have values for k_F and k_G that are derived in the same way as those for prior PA computations. We are now able to compare the values that we used in the computations from Section 4.2.1. The differences are shown in Table 4.8.

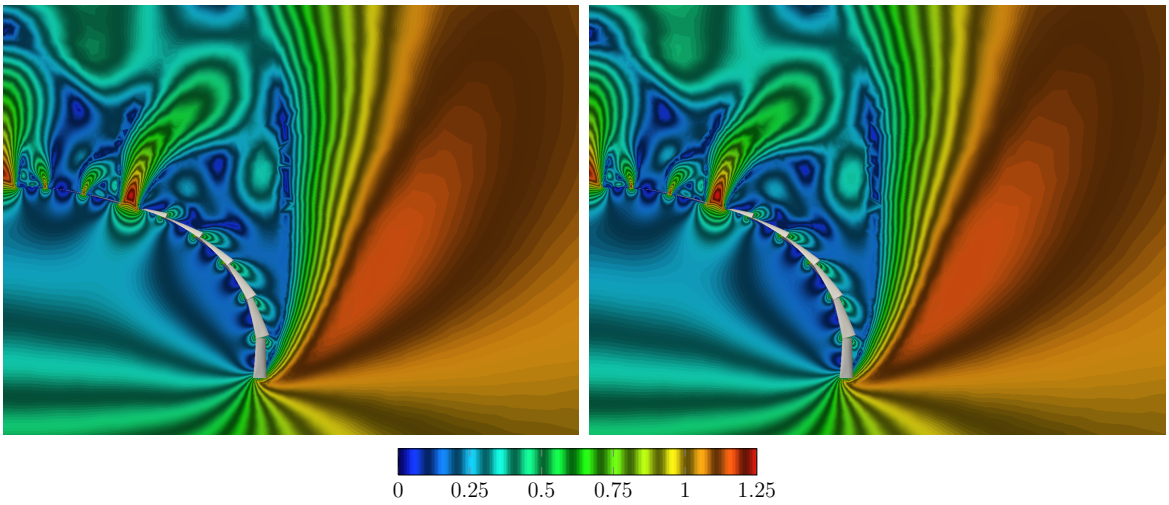


Figure 4.37: Normalized velocity magnitude (U/U_{ref}) near canopy shown on cut plane for full-scale (left) and subscale (right) JAXA parachute.

Table 4.8: Comparison between estimated values of k_F and k_G used in Section 4.2.1 and k_F and k_G calculated from the n-gore computation (note that Patches 1, 4, 5, and 11 do not have a k_G value because they do not contain a ring gap or sail slit). The k_F values are in CFM.

Patch	k_F (est)	k_F (cal)	% Diff	k_G (est)	k_G (cal)	% Diff
1	39.9	39.9	0.0			
2	39.9	39.9	0.0	0.970	0.900	7.2
3	39.9	39.8	0.3	0.970	0.902	7.0
4	39.9	39.8	0.3			
5	39.9	39.9	0.3			
6	39.9	39.8	0.3	0.831	0.664	20.1
7	39.9	39.8	0.3	0.831	0.654	21.3
8	39.9	39.8	0.3	0.831	0.648	22.0
9	39.9	39.8	0.3	0.831	0.638	23.2
10	39.9	39.8	0.3	0.831	0.655	21.2
11	39.9	39.8	0.3			

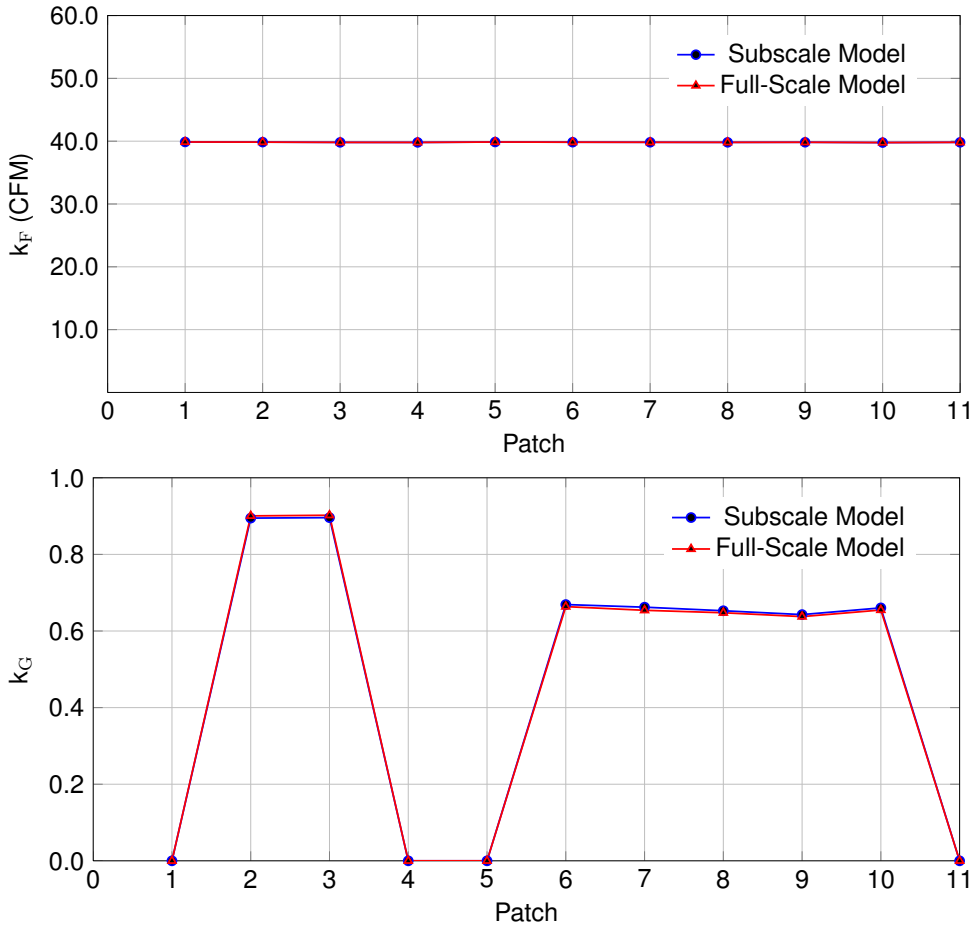


Figure 4.38: Calculated values for k_F and k_G for the 5-gore subscale and full-scale models.

Chapter 5

Concluding Remarks

The computational methods developed by the T★AFSM perform well in simulating a wide variety of parachute FSI problems. These methods help substantially increase the understanding of parachute dynamics and the effects of different designs, flight conditions, and configurations. They can reduce the cost of developing new spacecraft parachutes for new applications, as well as provide additional analysis that is difficult or impossible to acquire from empirical methods. Methods developed by the T★AFSM, including those presented in this thesis, continue to improve the accuracy of computational modeling of spacecraft parachutes and add to the value of computational methods in spacecraft parachute development. The rest of the concluding remarks here are from [59].

In this thesis we focused on FSI computation of clusters of ringsail spacecraft parachutes with modified geometric porosity. The modification, intended to increase the aerodynamic performance, consists of increasing the geometric porosity beyond the rings and sails with hundreds of ring gaps and sail slits by creating windows with removal of panels and wider gaps with removal of sails. This creates computational challenges beyond those created by the lightness of the canopy structure compared to the air masses involved in parachute dynamics, geometric complexities of hundreds of

gaps and slits, and the contact between the parachutes of the cluster or intra-canopy contact between the structural surfaces of a parachute. This is because the flow through the windows and wider gaps cannot be accurately modeled with the HMGP and needs to be actually resolved. All these computational challenges needed to be addressed simultaneously in FSI modeling reported in the thesis. This is accomplished with the SSTFSI technique, which serves as the core numerical technology, and a number of special FSI techniques. The special techniques include the newest version of the HMGP, which deals with the hundreds of gaps and slits, conservative version of the SENCT technique, which serves as a contact algorithm, and intra-canopy contact algorithms. In addition, we use two special techniques developed recently, one to remove the nonphysical spinning component of the parachute motion, and the other one to restore the mesh integrity lost during the mesh motion, but without resorting to remeshing. We presented results for 2- and 3-parachute clusters with two different payload models, as well as FSI results for a single subscale MP model.

Bibliography

[1] J. E. Akin, T. Tezduyar, M. Ungor, and S. Mittal. Stabilization parameters and Smagorinsky turbulence model. *Journal of Applied Mechanics*, 70:2–9, 2003.

[2] J. E. Akin and T. E. Tezduyar. Calculation of the advective limit of the SUPG stabilization parameter for linear and higher-order elements. *Computer Methods in Applied Mechanics and Engineering*, 193:1909–1922, 2004.

[3] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008.

[4] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. R. Hughes. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38:310–322, 2006.

[5] Y. Bazilevs, J. R. Gohean, T. J. R. Hughes, R. D. Moser, and Y. Zhang. Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198:3534–3550, 2009.

[6] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65:207–235, 2011.

[7] Y. Bazilevs, M.-C. Hsu, D. Benson, S. Sankaran, and A. Marsden. Computational fluid–structure interaction: Methods and application to a total cavopulmonary connection. *Computational Mechanics*, 45:77–89, 2009.

[8] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.

[9] Y. Bazilevs, M.-C. Hsu, K. Takizawa, and T. E. Tezduyar. ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 22(supp02):1230002, 2012.

[10] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. Isaksen. Computational fluid–structure interaction: Methods and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology*, 9:481–498, 2010.

[11] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J. Isaksen. A fully-coupled fluid–structure interaction simulation of cerebral aneurysms. *Computational Mechanics*, 46:3–16, 2010.

[12] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. Challenges and directions in computational fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 23:215–221, 2013.

[13] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, 2013.

[14] R. J. Benney, K. R. Stein, J. W. Leonard, and M. L. Accorsi. Current 3-D structural dynamic finite element modeling capabilities. In *Proceedings of AIAA*

14th Aerodynamic Decelerator Systems Technology Conference, AIAA Paper 97-1506, San Francisco, California, 1997.

[15] K.-U. Bletzinger, R. Wuchner, and A. Kupzok. Algorithmic treatment of shells and free form-membranes in FSI. In H.-J. Bungartz and M. Schafer, editors, *Fluid–Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 336–355. Springer, 2006.

[16] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[17] R. Calderer and A. Masud. A multiscale stabilized ALE formulation for incompressible flows with moving boundaries. *Computational Mechanics*, 46:185–197, 2010.

[18] A. Corsini, C. Iossa, F. Rispoli, and T. E. Tezduyar. A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors. *Computational Mechanics*, 46:159–167, 2010.

[19] A. Corsini, F. Rispoli, A. Santoriello, and T. E. Tezduyar. Improved discontinuity-capturing finite element techniques for reaction effects in turbulence computation. *Computational Mechanics*, 38:356–364, 2006.

[20] A. Corsini, F. Rispoli, and T. E. Tezduyar. Stabilized finite element computation of NOx emission in aero-engine combustors. *International Journal for Numerical Methods in Fluids*, 65:254–270, 2011.

[21] A. Corsini, F. Rispoli, and T. E. Tezduyar. Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique. *Journal of Applied Mechanics*, 79:010910, 2012.

[22] W. G. Dettmer and D. Peric. On the coupling between fluid flow and mesh motion in the modelling of fluid–structure interaction. *Computational Mechanics*, 43:81–90, 2008.

[23] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 5:283–292, 1977.

[24] M.-C. Hsu and Y. Bazilevs. Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations. *Finite Elements in Analysis and Design*, 47:593–599, 2011.

[25] M.-C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.

[26] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:329–349, 1981.

[27] A. A. Johnson and T. E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94, 1994.

[28] V. Kalro and T. E. Tezduyar. A parallel 3D computational method for fluid–structure interactions in parachute systems. *Computer Methods in Applied Mechanics and Engineering*, 190:321–332, 2000.

[29] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal of Scientific Computing*, 20:359–392, 1998.

[30] A. Lo. *Nonlinear Dynamic Analysis of Cable and Membrane Structure*. PhD thesis, Department of Civil Engineering, Oregon State University, 1982.

[31] R. Lohner, J. R. Cebral, C. Yang, J. D. Baum, E. L. Mestreau, and O. Soto. Extending the range of applicability of the loose coupling approach for FSI simulations. In H.-J. Bungartz and M. Schafer, editors, *Fluid–Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 82–100. Springer, 2006.

[32] C. J. Moorman. *Fluid–Structure Interaction Modeling of the Orion Spacecraft Parachutes*. PhD thesis, Rice University, 2010.

[33] S. Nagaoka, Y. Nakabayashi, G. Yagawa, and Y. J. Kim. Accurate fluid–structure interaction computations using elements without mid-side nodes. *Computational Mechanics*, 48:269–276, 2011.

[34] R. Ohayon. Reduced symmetric models for modal analysis of internal structural-acoustic and hydroelastic-sloshing systems. *Computer Methods in Applied Mechanics and Engineering*, 190:3009–3019, 2001.

[35] F. Rispoli, A. Corsini, and T. E. Tezduyar. Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (DCDD). *Computers & Fluids*, 36:121–126, 2007.

[36] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869, 1986.

[37] K. Stein, R. Benney, V. Kalro, T. E. Tezduyar, J. Leonard, and M. Accorsi. Parachute fluid–structure interactions: 3-D Computation. *Computer Methods in Applied Mechanics and Engineering*, 190:373–386, 2000.

[38] K. Stein, R. Benney, T. Tezduyar, and J. Potvin. Fluid–structure interactions of a cross parachute: Numerical simulation. *Computer Methods in Applied Mechanics and Engineering*, 191:673–687, 2001.

[39] K. Stein, T. Tezduyar, and R. Benney. Computational methods for modeling parachute systems. *Computing in Science and Engineering*, 5:39–46, 2003.

[40] K. Stein, T. Tezduyar, V. Kumar, S. Sathe, R. Benney, E. Thornburg, C. Kyle, and T. Nonoshita. Aerodynamic interactions between parachute canopies. *Journal of Applied Mechanics*, 70:50–57, 2003.

[41] K. R. Stein, R. J. Benney, V. Kalro, A. A. Johnson, and T. E. Tezduyar. Parallel computation of parachute fluid–structure interactions. In *Proceedings of AIAA 14th Aerodynamic Decelerator Systems Technology Conference*, AIAA Paper 97-1505, San Francisco, California, 1997.

[42] K. R. Stein, R. J. Benney, T. E. Tezduyar, J. W. Leonard, and M. L. Accorsi. Fluid–structure interactions of a round parachute: Modeling and simulation techniques. *Journal of Aircraft*, 38:800–808, 2001.

[43] K. Takizawa, Y. Bazilevs, and T. E. Tezduyar. Space–time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling. *Archives of Computational Methods in Engineering*, 19:171–225, 2012.

[44] K. Takizawa, J. Christopher, T. E. Tezduyar, and S. Sathe. Space–time finite element computation of arterial fluid–structure interactions with patient-specific data. *International Journal for Numerical Methods in Biomedical Engineering*, 26:101–116, 2010.

[45] K. Takizawa, M. Fritze, D. Montes, T. Spielman, and T. E. Tezduyar. Fluid–structure interaction modeling of ringsail parachutes with disreefing and modified geometric porosity. *Computational Mechanics*, 50:835–854, 2012.

[46] K. Takizawa, B. Henicke, A. Puntel, N. Kostov, and T. E. Tezduyar. Space–time techniques for computational aerodynamics modeling of flapping wings of an actual locust. *Computational Mechanics*, 50:743–760, 2012.

[47] K. Takizawa, B. Henicke, A. Puntel, N. Kostov, and T. E. Tezduyar. Computer modeling techniques for flapping-wing aerodynamics of a locust. *Computers & Fluids*, published online, DOI: 10.1016/j.compfluid.2012.11.008, November 2012.

[48] K. Takizawa, B. Henicke, A. Puntel, T. Spielman, and T. E. Tezduyar. Space–time computational techniques for the aerodynamics of flapping wings. *Journal of Applied Mechanics*, 79:010903, 2012.

[49] K. Takizawa, N. Kostov, A. Puntel, B. Henicke, and T. E. Tezduyar. Space–time computational analysis of bio-inspired flapping-wing aerodynamics of a micro aerial vehicle. *Computational Mechanics*, 50:761–778, 2012.

[50] K. Takizawa, D. Montes, M. Fritze, S. McIntyre, J. Boben, and T. E. Tezduyar. Methods for FSI modeling of spacecraft parachute dynamics and cover separation. *Mathematical Models and Methods in Applied Sciences*, 23:307–338, 2013.

[51] K. Takizawa, C. Moorman, S. Wright, J. Christopher, and T. E. Tezduyar. Wall shear stress calculations in space–time finite element computation of arterial fluid–structure interactions. *Computational Mechanics*, 46:31–41, 2010.

[52] K. Takizawa, C. Moorman, S. Wright, T. Spielman, and T. E. Tezduyar. Fluid–structure interaction modeling and performance analysis of the Orion spacecraft parachutes. *International Journal for Numerical Methods in Fluids*, 65:271–285, 2011.

[53] K. Takizawa, T. Spielman, C. Moorman, and T. E. Tezduyar. Fluid–structure interaction modeling of spacecraft parachutes for simulation-based design. *Journal of Applied Mechanics*, 79:010907, 2012.

[54] K. Takizawa, T. Spielman, and T. E. Tezduyar. Space–time FSI modeling and dynamical analysis of spacecraft parachutes and parachute clusters. *Computational Mechanics*, 48:345–364, 2011.

[55] K. Takizawa and T. E. Tezduyar. Multiscale space–time fluid–structure interaction techniques. *Computational Mechanics*, 48:247–267, 2011.

[56] K. Takizawa and T. E. Tezduyar. Bringing them down safely. *Mechanical Engineering*, 134(12):34–37, 2012.

[57] K. Takizawa and T. E. Tezduyar. Computational methods for parachute fluid–structure interactions. *Archives of Computational Methods in Engineering*, 19:125–169, 2012.

[58] K. Takizawa and T. E. Tezduyar. Space–time fluid–structure interaction methods. *Mathematical Models and Methods in Applied Sciences*, 22(supp02):1230001, 2012.

[59] K. Takizawa, T. E. Tezduyar, J. Boben, N. Kostov, C. Boswell, and A. Buscher. Fluid–structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity. *Computational Mechanics*, to be submitted, 2013.

[60] K. Takizawa, S. Wright, C. Moorman, and T. E. Tezduyar. Fluid–structure interaction modeling of parachute clusters. *International Journal for Numerical Methods in Fluids*, 65:286–307, 2011.

[61] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Parallel finite-element computation of 3D flows. *Computer*, 26(10):27–36, 1993.

[62] T. Tezduyar and Y. Osawa. Fluid–structure interactions of a parachute crossing the far wake of an aircraft. *Computer Methods in Applied Mechanics and Engineering*, 191:717–726, 2001.

[63] T. E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28:1–44, 1992.

[64] T. E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8:83–130, 2001.

[65] T. E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43:555–575, 2003.

[66] T. E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. In E. Stein, R. D. Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, chapter 17. John Wiley & Sons, 2004.

[67] T. E. Tezduyar. Finite elements in fluids: Special methods and enhanced solution techniques. *Computers & Fluids*, 36:207–223, 2007.

[68] T. E. Tezduyar. Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces. *Computers & Fluids*, 36:191–206, 2007.

[69] T. E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94(3):339–351, 1992.

[70] T. E. Tezduyar, M. Behr, S. Mittal, and A. A. Johnson. Computation of unsteady incompressible flows with the finite element methods – space–time formulations, iterative strategies and massively parallel implementations. In *New Methods in*

Transient Analysis, PVP-Vol.246/AMD-Vol.143, pages 7–24, New York, 1992. ASME.

[71] T. E. Tezduyar, M. Behr, S. Mittal, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, 94(3):353–371, 1992.

[72] T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.

[73] T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[74] T. E. Tezduyar and S. Sathe. Enhanced-discretization selective stabilization procedure (EDSSP). *Computational Mechanics*, 38:456–468, 2006.

[75] T. E. Tezduyar and S. Sathe. Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques. *International Journal for Numerical Methods in Fluids*, 54:855–900, 2007.

[76] T. E. Tezduyar, S. Sathe, T. Cragin, B. Nanna, B. S. Conklin, J. Pausewang, and M. Schwaab. Modeling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics. *International Journal for Numerical Methods in Fluids*, 54:901–922, 2007.

[77] T. E. Tezduyar, S. Sathe, R. Keedy, and K. Stein. Space–time finite element techniques for computation of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 195:2002–2027, 2006.

[78] T. E. Tezduyar, S. Sathe, J. Pausewang, M. Schwaab, J. Christopher, and J. Crabtree. Interface projection techniques for fluid–structure interaction modeling with moving-mesh methods. *Computational Mechanics*, 43:39–49, 2008.

[79] T. E. Tezduyar, S. Sathe, M. Schwaab, J. Pausewang, J. Christopher, and J. Crabtree. Fluid–structure interaction modeling of ringsail parachutes. *Computational Mechanics*, 43:133–142, 2008.

[80] T. E. Tezduyar, S. Sathe, and K. Stein. Solution techniques for the fully-discretized equations in computation of fluid–structure interactions with the space–time formulations. *Computer Methods in Applied Mechanics and Engineering*, 195:5743–5753, 2006.

[81] T. E. Tezduyar, M. Senga, and D. Vicker. Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing. *Computational Mechanics*, 38:469–481, 2006.

[82] T. E. Tezduyar, K. Takizawa, T. Brummer, and P. R. Chen. Space–time fluid–structure interaction modeling of patient-specific cerebral aneurysms. *International Journal for Numerical Methods in Biomedical Engineering*, 27:1665–1710, 2011.

[83] T. E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher. Space–time finite element computation of complex fluid–structure interactions. *International Journal for Numerical Methods in Fluids*, 64:1201–1218, 2010.

[84] E. H. van Brummelen and R. de Borst. On the nonnormality of subiteration for a fluid-structure interaction problem. *SIAM Journal on Scientific Computing*, 27:599–621, 2005.

[85] S. E. Wright. *Advanced Fluid–Structure Interaction Techniques for Modeling Ringsail Parachutes*. PhD thesis, Rice University, 2010.